



华清远见系列图书

深入浅出，依靠深厚行业经验讲透技术原理
循序渐进，详解典型应用案例提升实战能力



超大容量多媒体，总时长超过29小时

QQ答疑：454178091

Ubuntu Linux

从初学到精通

◎ 华清远见嵌入式培训中心 潘光洋 等编著

- ◆ 提供全书内容的多媒体教学视频635分钟
- ◆ 提供VMware工具、Linux命令工具、编辑器工具、GCC工具、GDB工具、Shell工具、make工具、Eclipse开发工具、kdevelop开发工具及项目管理Subversion工具等Linux常用工具教学视频581分钟
- ◆ 提供209个常用linux命令教学视频583分钟

Ubuntu Linux从初学到精通

· 学员评价 ·

学习过程中总会遇到一些问题，通过翻书很快帮我找到了答案，进步很快。

安普盛科技 郝工

谢谢各位网友给我推荐的几本好书，也特别感谢华清老师的帮助和推荐，让我从Ubuntu的门外汉逐步开始蜕变。

网友 不断进步中

- ◆ Linux 概述
- ◆ Ubuntu Linux概述
- ◆ 安装前的准备
- ◆ 单系统安装Ubuntu
- ◆ Windows下安装Ubuntu
- ◆ 快速配置系统
- ◆ GNOME桌面环境
- ◆ KDE桌面环境
- ◆ Xfce桌面环境
- ◆ Shell基础应用
- ◆ 软件包管理
- ◆ 常用应用软件
- ◆ 文件系统管理
- ◆ 系统管理
- ◆ 网络管理
- ◆ 网络应用服务器
- ◆ 文件服务器
- ◆ Web服务器
- ◆ 编程工具
- ◆ Shell编程
- ◆ 版本控制工具



责任编辑：胡辛征
封面设计：侯士卿

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书。



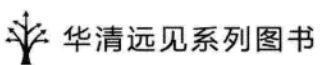
上架建议：操作系统 > Linux

ISBN 978-7-121-12813-4



9 787121 128134 >

定价：59.00元（含DVD光盘1张）



Ubuntu Linux

从初学到精通

◎ 华清远见嵌入式培训中心 潘光洋 等编著

电子工业出版社
Publishing House of Electronics Industry
北京•BEIJING



内 容 简 介

Ubuntu Linux 是一个操作系统，其提供给用户一个操作简单、使用方便的桌面环境。本书详述了 Ubuntu Linux 的安装和使用。其中内容包括了 Linux 操作系统的概述、Ubuntu 系列操作系统的安装、Ubuntu 的多种桌面环境、Ubuntu 的命令操作、Ubuntu 中常用软件的安装与使用、Ubuntu 服务器的应用，以及 Ubuntu 环境下的编程。为使读者能快速掌握 Ubuntu 系统的操作，本书结合相关步骤的图例详细讲解了每个操作步骤。

本书配套光盘中提供了源文件和操作步骤的视频教程。视频配合音频能帮助读者理解和掌握操作。

本书非常适合刚使用 Ubuntu Linux 操作系统的普通用户和系统管理员，同时也适合以 Ubuntu 为开发环境的开发者。

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

Ubuntu Linux 从初学到精通/潘光洋等编著. —北京：电子工业出版社，2011.4

(华清远见系列图书)

ISBN 978-7-121-12813-4

I . ①U… II . ①潘… III. ①Linux 操作系统 IV. ①TP316.89

中国版本图书馆 CIP 数据核字 (2011) 第 010275 号

责任编辑：胡辛征

印 刷：三河市鑫金马印装有限公司
装 订：

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：31 字数：794 千字

印 次：2011 年 4 月第 1 次印刷

印 数：4000 册 定价：59.00 元（含 DVD 光盘 1 张）

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

随着 Linux 操作系统的不断发展，Linux 集成了图形操作界面、简化了用户的操作，使得越来越多的用户使用 Linux 操作系统。Ubuntu Linux 是众多 Linux 系统发行版本中一款优秀的图形操作界面的操作系统。Ubuntu Linux 是开源免费的，相当一部分计算机用户也从 Windows 的操作系统中转为使用 Ubuntu Linux 系统。Ubuntu 中拥有与 Windows 操作系统常用软件的功能相似的软件。

为推广 Linux 的使用，以及使广大用户能够更方便地掌握和使用 Ubuntu Linux 系统，作者编写了此书。希望读者学习本书后能够自行完成 Ubuntu 的安装，掌握 Ubuntu 中常用软件的操作和构建 Ubuntu 服务器的方法，以及能够在 Ubuntu 环境下进行软件开发。

本书内容充实，实例多，并提供了 9.5 小时的音频、视频教程，完整地讲述了 Ubuntu Linux 的安装和操作，使读者能够结合视频教程来完成学习过程。

本书共 22 章，分 7 部分。

1) 第 1 部分：走进 Linux

这一部分包括了第 1~2 章，主要描述了 Linux 的发展、内核和发行；讲述 Ubuntu Linux 的发展历程，以及它的特点。这部分让读者了解 Linux 和 Ubuntu Linux 的基础知识，加深对 Linux 的认识。

2) 第 2 部分：Linux 的安装

这一部分包括了第 3~6 章，主要讲述 Ubuntu 安装的多种方式。第 3 章讲述安装 Ubuntu 之前的准备工作；第 4 章讲述 Ubuntu 单独安装的方法和步骤；第 5 章讲述在 Windows 系统下安装 Ubuntu 的 3 种方法，包括使用 Wubi 进行无痛安装、使用 GRUB4DOS 进行安装和使用虚拟机进行安装；第 6 章讲述安装 Ubuntu 系统后对系统进行配置，使得系统适合个人的使用习惯。通过这部分的学习，读者可掌握安装 Ubuntu 系统的方法和技巧，掌握安装 Ubuntu 后配置系统的方法。本部分的操作过程都以多媒体视频的方式呈现，读者能够按照操作视频一步一步地完成安装操作。

3) 第 3 部分：Ubuntu 的桌面环境

这一部分包括第 7~9 章，主要讲述 Ubuntu 系列中的 3 个环境桌面系统，包括 GNOME 桌面环境、KDE 桌面环境和 Xfce 桌面环境。第 7 章讲述 GNOME 桌面环境的起源和发展，讲述 GNOME 桌面下面板、桌面和首选项的操作，介绍 GNOME 桌面下常用应用软件；第 8 章讲述 KDE 桌面环境的起源和发展，讲述 Kubuntu 的概念，描述在 Ubuntu 中安装 KDE 桌面环境的步骤，讲述 KDE 桌面下面板、桌面和首选项的操作，介绍 KDE 桌面下常用应用软件；第 9 章讲述 Xfce 桌面环境的起源和发展，讲述 Xubuntu 的基础知识，讲述在 Ubuntu 下安装 Xfce 桌面的步骤，讲述 Xfce 桌面下面板、桌面和首选项的操作，介绍了 Xfce 桌面下常用应用软件。通过这部分的学习，读者能了解 Ubuntu 系列各个桌面环境。

4) 第 4 部分：基本应用

这一部分包括第 10~12 章，主要讲述 shell 命令的使用、软件包的安装和常用软件的使用。第 10 章讲述 shell 的特点、命令行的使用方法和技巧，以及系统命令的使用；第 11 章讲述多种软件包管理工具，包括 dpkg、apt-get、Aptitude 和新立得软件包管理器；第 12 章讲述 Ubuntu

下常用的应用软件，包括 Gedit 文本编辑器、OpenOffice.org 办公套件、照片管理器 F-Spot、图像处理软件 GIMP、电影播放机 Totem、音乐播放器 Rhythmbox、刻录软件 GNOMEBaker、腾讯 QQ、互联网通信 Pidgin、飞信 LibFetion、下载工具 gFTP、下载工具电驴 aMule、浏览器 Firefox、内核全虚拟化 KVM、桌面虚拟机 VirtualBox OSE 和电子邮件管理程序 Evolution。完成这部分的学习，读者能够方便地使用 Ubuntu 进行日常的学习、娱乐和工作。

5) 第 5 部分：高级应用

这一部分包括第 13~15 章，主要讲述文件系统管理、系统管理和网络管理。第 13 章讲述 Linux 文件系统的概念和类型，讲述 Ubuntu 下的磁盘分区工具 fdisk 和 parted，讲述文件系统的操作命令和磁盘配额操作；第 14 章讲述系统的用户管理、组管理、进程管理、系统备份和还原，同时讲述了系统信息的查看方法；第 15 章讲述 Linux 网络的基本原理，讲述网络工具 GNOME-nettool 的使用，讲述 Ubuntu 连接网络的配置。这部分内容使读者能更深入学习 Ubuntu 操作系统的知识。

6) 第 6 部分：服务器

这一部分包括第 16~19 章，主要介绍 Ubuntu 系统下的服务器，包括域名服务器 Bind9、文件传输服务器 vsFTPD、电子邮件服务器 Postfix、文件服务器 Samba 与 NFS、Web 网络服务器 Apache 和数据库服务器 MySQL。第 16 章讲述域名服务器 Bind9、文件传输服务器 vsFTPD 和电子邮件服务器 Postfix 的安装、配置和应用；第 17 章讲述文件服务器 Samba 与 NFS 的安装、配置和应用；第 18 章讲述 Web 网络服务器 Apache 的安装、配置和应用，讲述了部署 PHP 站点的方法；第 19 章讲述 MySQL 数据库在 Ubuntu 中的安装、配置和应用。本部分服务器的使用要求对 Linux 有一定了解的系统管理员来完成。每个服务器的应用都有一个相应简单的实例，系统管理员能够通过实例掌握各个服务器构建的方法和技巧。

7) 第 7 部分：Linux 编程

这一部分包括第 20~22 章，主要讲述 Ubuntu Linux 环境下 C/C++ 的编程环境、shell 编程和版本控制。第 20 章讲述 Ubuntu 的编程概述，讲述使用 Ubuntu 下的集成开发环境，以及常用命令工具编写 C/C++ 程序；第 21 章讲述 shell 的语法和编写 shell 脚本；第 22 章讲述构建 subversion 版本控制服务器和使用 subversion 客户端管理程序代码的方法。

本书适合的读者包括 Linux 的爱好者、普通用户、Ubuntu 系统管理员、Linux 程序员和计算机专业的大、中专学生等。普通用户最好的阅读方式是按顺序阅读前 4 部分的内容，后 3 部分可作为选读的内容。Ubuntu 系统管理员和 Linux 程序员对书上的内容可选读，选择需要的部分进行阅读。

本书光盘中的视频教程与书中有图示的例子相对应，服务器的构建和配置都有视频教程。

本书由潘光洋编写，此外，高淑娟、王丽娜、周毅、林小峰、刘刚、马海波、李强、吴慧、马玉刚、冯浩、唐爱琴、李子龙、王明明、蒋志等同志也参与了本书的编写工作。在此，编者对他们表示衷心的感谢。

由于时间仓促，加之水平有限，故本书中难免存在疏漏、欠妥，甚至错误之处，敬请读者批评指正，也恳切期望大家提出宝贵意见。

编著者

目 录

第 1 部分 走进 Linux

第 1 章 Linux 概述.....	1
1.1 Linux 的发展.....	2
1.1.1 UNIX 的发展	2
1.1.2 MINIX 系统	3
1.1.3 POSIX 标准	3
1.1.4 Linux 的诞生	4
1.1.5 Linux 的命名由来和发音	6
1.1.6 Linux 的标志	6
1.2 Linux 的内核和发行.....	6
1.2.1 内核的发展	7
1.2.2 主流发行版本	7
1.2.3 GNU 通用公共许可证	9
1.3 小结	9
1.4 习题	10

第 2 章 Ubuntu Linux 概述.....	11
2.1 Ubuntu 简史	12
2.1.1 Debian 的历史	12
2.1.2 Ubuntu 的发展	13
2.2 Ubuntu 的特点	14
2.3 Ubuntu 的发行	14
2.4 Ubuntu 9.04 新功能	16
2.5 小结	16
2.6 习题	17

第 2 部分 Linux 的安装

第 3 章 安装前的准备.....	18
3.1 硬件需求	19

3.1.1	CPU	20
3.1.2	硬盘	20
3.1.3	内存	20
3.1.4	显示卡	20
3.1.5	光驱	21
3.1.6	网卡	21
3.1.7	键盘与鼠标	21
3.2	软件需求	21
3.2.1	软件下载	21
3.2.2	光盘刻录	22
3.3	安装策略	23
3.3.1	记录硬件信息	23
3.3.2	安装方式的选择	23
3.3.3	规划磁盘分区	24
3.4	注意事项	26
3.4.1	下载映像文件	26
3.4.2	硬盘分区	26
3.4.3	网络配置	27
3.5	小结	27
3.6	习题	27
	第4章 单系统安装 Ubuntu	28
4.1	启动 Live CD	29
4.2	在 Live CD 中磁盘分区	31
4.2.1	运行 Partition Editor	31
4.2.2	划分磁盘分区	33
4.3	安装系统步骤	36
4.3.1	选择语言	36
4.3.2	选择时区	37
4.3.3	键盘布局	37
4.3.4	预备硬盘空间	37
4.3.5	准备分区	38
4.3.6	创建用户	39
4.3.7	迁移向导	40
4.3.8	安装系统配置清单	40
4.4	小结	41
4.5	习题	41

第 5 章 Windows 下安装 Ubuntu	42
5.1 Wubi 无痛安装.....	43
5.1.1 Wubi 安装的特点.....	43
5.1.2 Wubi 安装配置.....	43
5.1.3 Wubi 安装过程.....	45
5.1.4 Wubi 卸装 Ubuntu.....	46
5.2 使用 Grub4dos 进行安装.....	47
5.2.1 GRU4COX 简介.....	47
5.2.2 GRUB4DOS 安装.....	47
5.2.3 GRUB4DOS 安装 Ubuntu	48
5.3 虚拟机安装 Ubuntu	48
5.3.1 虚拟机的安装	49
5.3.2 虚拟环境的配置.....	49
5.3.3 安装过程	50
5.4 小结	51
5.5 习题	51
第 6 章 快速配置系统.....	52
6.1 语言支持设置	53
6.2 设置 SCIM	54
6.2.1 设置 SCIM 开关键热键.....	55
6.2.2 禁止部分输入法引擎.....	56
6.2.3 设置垂直式候选词表.....	57
6.3 设置日期时间	58
6.4 设置屏幕保护	59
6.5 设置外观	59
6.5.1 修改主题	60
6.5.2 自定义主题	60
6.5.3 安装新主题	61
6.5.4 设置桌面背景	62
6.5.5 开启三维显示效果.....	63
6.6 设置分辨率	63
6.7 设置登录窗口	64
6.7.1 设置本地登录窗口.....	64
6.7.2 设置自动登录	65
6.8 设置软件源	65
6.8.1 设置软件源服务器.....	66
6.8.2 设置软件更新	66

6.9 启动程序	67
6.10 网络链接	68
6.10.1 设置静态 IP 地址	68
6.10.2 设置 ADSL 网络连接	69
6.11 小结	70
6.12 习题	70

第 3 部分 Ubuntu 的桌面环境

第 7 章 GNOME 桌面环境	71
7.1 GNOME 的介绍	72
7.2 GNOME 的面板和桌面	73
7.2.1 GNOME 面板	73
7.2.2 面板操作	75
7.2.3 GNOME 桌面	77
7.3 GNOME 首选项	78
7.3.1 主菜单	79
7.3.2 首选应用程序	79
7.3.3 键盘快捷键	80
7.4 GNOME 应用软件	81
7.4.1 字符终端	82
7.4.2 文本编辑器	82
7.4.3 计算器	82
7.4.4 图像编辑软件	83
7.4.5 GNOME 媒体播放器	83
7.4.6 刻录软件 Brasero	84
7.4.7 邮件程序 Evolution	85
7.4.8 游戏	85
7.5 小结	86
7.6 习题	86
第 8 章 KDE 桌面环境	87
8.1 KDE 简介	88
8.1.1 KDE 起源	88
8.1.2 KDE 的发展	89
8.1.3 KDE 前进动力和保障	89
8.2 Kubuntu 的简介	90
8.3 Ubuntu 中安装 KDE 桌面	91

8.3.1 新立得安装 KDE 桌面	91
8.3.2 命令行安装 KDE	94
8.4 KDE 的面板和桌面	94
8.4.1 面板	94
8.4.2 操作面板和桌面	96
8.5 KDE 桌面环境的配置	98
8.5.1 修改时间日期	98
8.5.2 修改显示器的分辨率	99
8.5.3 设置软件源与更新软件	100
8.5.4 设置默认的应用程序	100
8.6 KDE 桌面环境中常用软件	101
8.6.1 Konqueror 网页浏览器	101
8.6.2 Dolphin 文件管理器	101
8.6.3 K3b 烧录软件	102
8.6.4 Kate 高级文本编辑器	102
8.6.5 KMail 电子邮件客户端	103
8.6.6 Konsole 模拟终端	103
8.6.7 KPackageKit 软件包管理器	103
8.6.8 System Monitor	104
8.7 小结	105
8.8 习题	105
第 9 章 Xfce 桌面环境	106
9.1 Xfce 简介	107
9.1.1 Xfce 简史	107
9.1.2 Xfce 的精神	107
9.2 Xubuntu 的简介	108
9.3 在 Ubuntu 中安装 Xfce	109
9.4 Xfce 面板和桌面	111
9.4.1 面板	111
9.4.2 桌面	113
9.4.3 操作面板和桌面	113
9.5 Xfce 的系统设置	116
9.5.1 设置显示器分辨率和刷新率	116
9.5.2 设置首选应用程序	116
9.5.3 设置工作区桌面的名称	117
9.6 Xfce 的组件和常用应用程序	117
9.6.1 Xfce4 的核心组件	117

9.6.2 应用程序	118
9.7 小结	120
9.8 习题	120

第 4 部分 基本应用

第 10 章 shell 基础应用	121
10.1 认识 shell	122
10.1.1 shell 的概念	122
10.1.2 shell 的类型	123
10.1.3 shell 的特点	124
10.1.4 命令行语法	125
10.2 标准输入输出	126
10.2.1 重定向	127
10.2.2 管道	129
10.3 通配符	130
10.3.1 通配符?	130
10.3.2 通配符*	131
10.3.3 字符范围符[]	131
10.4 shell 操作	132
10.4.1 启动终端	132
10.4.2 更换 shell	133
10.4.3 配置 shell	134
10.4.4 编辑命令行	137
10.4.5 命令行补全	138
10.4.6 后台执行	138
10.4.7 退出 shell	140
10.5 常用的 Linux 命令	140
10.5.1 基本操作命令	140
10.5.2 帮助命令	143
10.5.3 文件和目录操作命令	146
10.5.4 文档打包和压缩命令	147
10.5.5 系统操作命令	149
10.6 小结	150
10.7 习题	150

第 11 章 软件包管理	151
11.1 软件包的概述	152
11.1.1 DEB 软件包	152
11.1.2 RPM 软件包	156
11.1.3 源码包	157
11.1.4 虚拟软件包	158
11.2 字符界面软件包管理工具	158
11.2.1 dpkg 软件包管理工具	158
11.2.2 apt-get 软件包管理工具	164
11.2.3 Aptitude 软件包管理工具	170
11.2.4 其他软件包管理工具	174
11.3 新立得软件包管理工具	176
11.3.1 启动新立得管理工具	176
11.3.2 设置软件源	177
11.3.3 查找软件包	178
11.3.4 安装软件包	180
11.3.5 卸载软件包	181
11.4 小结	182
11.5 习题	182
第 12 章 常用应用软件	183
12.1 文本编辑工具 Gedit	184
12.1.1 启动 Gedit	184
12.1.2 处理文件	185
12.1.3 处理文本	188
12.1.4 高亮显示	189
12.2 办公软件包 OpenOffice.org	190
12.2.1 启动 OpenOffice.org 套件	190
12.2.2 OpenOffice.org Writer	191
12.2.3 Calc	199
12.2.4 Impress	204
12.3 多媒体软件	208
12.3.1 照片管理器 F-Spot	208
12.3.2 图片处理器 GIMP	211
12.3.3 电影播放机 Totem	215
12.3.4 音乐播放器 Rhythmbox	216
12.3.5 刻录软件 GNOME Baker	217
12.4 即时通信	219

12.4.1 腾讯 QQ	219
12.4.2 互联网通信 Pidgin	220
12.4.3 飞信 LibFetion	221
12.5 下载工具	221
12.5.1 gFTP	222
12.5.2 Linux 下的电驴 aMule	223
12.6 浏览器 Firefox	226
12.7 虚拟机软件	230
12.7.1 内核全虚拟化 KVM	230
12.7.2 桌面虚拟机 VirtualBox OSE	232
12.8 电子邮件 Evolution	236
12.9 小结	244
12.10 习题	244

第 5 部分 高级应用

第 13 章 文件系统管理	245
13.1 文件系统概述	246
13.2 Linux 支持的主要文件系统	246
13.2.1 ext2 文件系统	247
13.2.2 ext3 文件系统	247
13.2.3 ext4 文件系统	248
13.2.4 ReiserFs 文件系统	249
13.2.5 VFS 文件系统	249
13.2.6 iso9660 文件系统	250
13.2.7 其他文件系统	250
13.3 Linux 文件系统目录结构	250
13.3.1 文件概念	251
13.3.2 目录结构	252
13.4 磁盘分区工具	253
13.4.1 fdisk 分区工具	254
13.4.2 parted 分区工具	258
13.5 文件系统管理基本命令	260
13.5.1 dd 命令——磁盘备份和数据转换	260
13.5.2 df 命令——检查文件系统占用空间情况	261
13.5.3 du 命令——显示文件占用磁盘空间情况	262
13.5.4 挂载命令 mount	263
13.5.5 卸载命令 umount	265

13.6 磁盘配额	266
13.6.1 磁盘配额的简介	266
13.6.2 配置磁盘配额	266
13.7 小结	269
13.8 习题	269
第 14 章 系统管理	270
14.1 用户管理	271
14.1.1 创建 SUID/SGID 用户账户	271
14.1.2 删除用户	272
14.1.3 修改用户属性	273
14.1.4 passwd 和 shadow 文件	273
14.1.5 常用命令	275
14.2 组管理	278
14.2.1 用户组的添加和删除	278
14.2.2 修改组的属性	279
14.2.3 /etc/group 文件	280
14.3 进程管理	280
14.3.1 进程的概念	280
14.3.2 进程显示和删除	281
14.3.3 进程调度	284
14.3.4 系统监视器管理进程	287
14.4 查看系统信息	288
14.4.1 系统硬件信息	288
14.4.2 系统日志	290
14.4.3 模块加载信息	290
14.5 系统备份与还原	291
14.5.1 备份恢复的概念	291
14.5.2 备份恢复的策略	292
14.5.3 tar 命令备份和还原/home 目录	292
14.5.4 dump 备份和 restore 还原文件	293
14.6 小结	294
14.7 习题	294
第 15 章 网络管理	296
15.1 Linux 网络基础	297
15.1.1 计算机网络体系基本概念	297
15.1.2 TCP/IP 协议族	299
15.1.3 TCP 协议	301
15.1.4 UDP 协议	303

15.1.5 IP 协议	304
15.2 网络工具 GNOME-nettool.....	306
15.2.1 查看本地 IP 地址	306
15.2.2 验证与远程计算机的连接	307
15.2.3 查看路由表信息	307
15.2.4 扫描开放的端口	308
15.3 网络配置	309
15.3.1 网络配置文件	309
15.3.2 网络常用命令	312
15.4 小结	317
15.5 习题	317

第 6 部分 服务器

第 16 章 网络应用服务器	318
16.1 域名服务器 Bind9.....	319
16.1.1 域名系统 DNS 简介	319
16.1.2 Bind 服务器简介	321
16.1.3 安装及启动 Bind9	321
16.1.4 配置 Bind9 服务器	321
16.1.5 测试 Bind9 服务器	326
16.2 文件传输服务器 vsFTPD	329
16.2.1 FTP 服务器的特性	329
16.2.2 FTP 用户类型	330
16.2.3 安装 vsFTPD 服务器，以及它的基本操作	330
16.2.4 配置 vsFTPD 服务器	331
16.2.5 配置本地组访问的 FTP	333
16.3 电子邮件服务器 Postfix	335
16.3.1 Postfix 简介	335
16.3.2 Postfix 的总体结构	336
16.3.3 安装并运行 Postfix	338
16.3.4 配置 Postfix	339
16.3.5 测试 Postfix 服务器	342
16.4 小结	345
16.5 习题	345
第 17 章 文件服务器	346
17.1 Samba 服务器概述	347
17.1.1 SMB 协议简介	347
17.1.2 Samba 简介	347

17.1.3 Samba 服务器安装和启动	349
17.2 Samba 的配置	350
17.2.1 smb.conf 配置文件的组成	350
17.2.2 smb.conf 的编写格式和语法	350
17.2.3 smb.conf 全局设置	351
17.2.4 smb.conf 共享设置	352
17.3 使用 Samba 共享实例	353
17.3.1 设置用户的 Samba 密码	353
17.3.2 使用配置文件设置共享资源	354
17.3.3 使用图形操作界面设置共享	355
17.3.4 使用 Samba 的客户端	356
17.4 NFS 文件服务器	359
17.4.1 NFS 服务简介	360
17.4.2 安装和启动 NFS 服务器	360
17.4.3 配置 NFS 服务器	361
17.4.4 客户端 NFS 的链接	363
17.5 小结	365
17.6 习题	365
第 18 章 Web 服务器	367
18.1 Web 服务器简介	368
18.1.1 Apache 服务器	368
18.1.2 Tomcat 服务器	369
18.1.3 其他 Web 服务器	370
18.2 构建 Apache 服务器	371
18.2.1 安装 Apache 服务器	371
18.2.2 Apache 配置文件	371
18.2.3 Apache 的基本配置	374
18.3 管理 Apache 服务器	376
18.4 Apache 日志管理	377
18.4.1 日志相关配置指令	377
18.4.2 访问日志	377
18.4.3 错误日志	379
18.5 配置动态 Web 站点	380
18.6 小结	381
18.7 习题	382
第 19 章 数据库 MySQL	383
19.1 数据库简介	384
19.1.1 MySQL 数据库	384
19.1.2 PostgreSQL 数据库	384

19.2 构建 MySQL 数据库服务器.....	385
19.2.1 安装 MySQL 数据库服务器	385
19.2.2 配置 MySQL 数据库服务器	386
19.2.3 启动和关闭 MySQL 数据库服务器	390
19.3 使用 MySQL 数据库	391
19.3.1 运行 MySQL 命令	391
19.3.2 数据库的管理	392
19.3.3 表的管理.....	394
19.3.4 数据记录的管理	400
19.3.5 数据库用户管理	404
19.3.6 数据库的备份和恢复	406
19.4 小结	407
19.5 习题	408

第 7 部分 Linux 编程

第 20 章 编程工具	409
20.1 Ubuntu 下编程环境概述.....	410
20.1.1 编程工具简介	410
20.1.2 Ubuntu 下的编程类型	410
20.2 常用编辑器	411
20.2.1 VIM 编辑器	411
20.2.2 Emacs 编辑器	416
20.3 集成开发环境.....	418
20.3.1 Anjuta	419
20.3.2 Eclipse	421
20.4 C/C++程序开发工具.....	425
20.4.1 编译器工具 GCC.....	425
20.4.2 调试工具 gdb.....	427
20.4.3 make 工具	429
20.5 小结	431
20.6 习题	431
第 21 章 shell 编程	433
21.1 Bash 简介	434
21.1.1 Bash 的优势	434
21.1.2 Bash 脚本的编写与运行	434
21.2 Bash 的语法	435
21.2.1 shell 变量	435

21.2.2 条件语句	439
21.2.3 循环语句	442
21.2.4 循环控制	445
21.2.5 case 测试语句	447
21.2.6 select 循环语句	448
21.2.7 shell 内置命令	449
21.3 shell 函数	451
21.3.1 函数简介	451
21.3.2 函数的参数和返回值	452
21.4 shell 脚本综合实例	454
21.5 小结	457
21.6 习题	457
第 22 章 版本控制工具	459
22.1 版本控制简介	460
22.1.1 CVS 简介	460
22.1.2 SVN 简介	461
22.2 构建 svn 服务器	461
22.2.1 安装 Subversion	462
22.2.2 创建 svn 数据仓库	462
22.2.3 配置 apache 的 svn 模块	465
22.2.4 配置用户权限	466
22.3 svn 字符界面客户端	469
22.3.1 检出文件	469
22.3.2 添加文件	470
22.3.3 提交文件	470
22.3.4 更新文件	471
22.3.5 查看日志	472
22.4 svn 图形界面客户端 esvn	473
22.4.1 安装 esvn	473
22.4.2 配置 esvn	473
22.4.3 检出文件	474
22.4.4 打开工作副本	475
22.4.5 提交文件	475
22.4.6 更新工作副本	476
22.4.7 查看日志	476
22.5 小结	477
22.6 习题	477

Linux

第1部分 走进 Linux

第1章

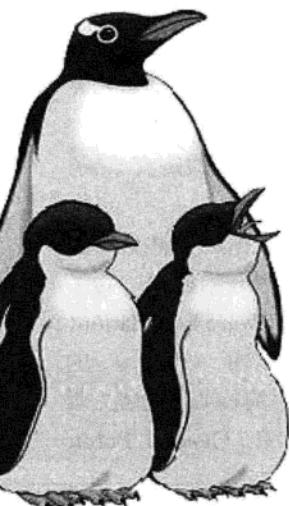
Linux 概述

也许你使用过其他 Linux 的发行版本，也许你才开始接触 Linux。本章主要介绍有关 Linux 的基础知识。从 Linux 的来历，到它的发展来简述 Linux。

如果你刚刚开始结束对 Linux 的学习，我觉得你还有必要知道什么是 Linux 及了解它的发展，了解 GNU。在这章中，你将对 Linux 有一个全新的认识。Linux 是一个与你之前使用的 Windows 完全不一样的操作系统。

如果你已经使用过 Linux 的其他发行版本，那么也来请你阅读一下本书，以巩固对 Linux 的认识，增加对 Linux 的了解。

现在，我们就开始走进奇妙的 Linux 世界，去揭开它的神秘面纱。





1.1 Linux 的发展



每一件事物都有它自己的由来，Linux 也是如此，有着一段不平凡的故事。接下来，笔者就从 Linux 的诞生和发展来给大家简述它的故事。

说到 Linux，我们就得从 UNIX 开始说起。在 Linux 还没有出现之前，一个稳定而成熟的操作系统 UNIX 就存在已久了。那为什么要说 UNIX 呢？它们之间有什么关系呢？让笔者慢慢给大家讲解，看完之后，你会对 UNIX 和 Linux 有一个大致的了解。

1.1.1 UNIX 的发展

大家都知道只有裸机是不可以工作的，唯有在裸机上安装了操作系统，它才可以运行，为人们服务。早期计算机可以说是一种奢侈品，很少有人能买得起，只有在军事上和一些比较大的研究所才能拥有一台体型庞大的大型机（计算处理能力很低，不能与现在相比较）。一个研究所只能公用一台计算机，那就要求操作系统可以支持多人使用，放在 19 世纪 60 年代初期，麻省理工学院（MIT）开发了所谓的“兼容分时系统（Compatible Time-Sharing System, CTSS）”，这个系统可以让约 30 人同时使用计算机。为了加强 CTSS 的分布式性能，在 1965 年前后由贝尔研究室（Bell）、麻省理工学院（MIT）及奇异电器公司（GE）开始来共同开发一个名为 Multics 的大型计划，目标是想让大型主机可以联机 300 台以上的使用者。往往计划过大，失败就成为必然。Multics 是一个没有完成的计划。计划失败之后，组织解散。

而参与过 Multics 计划的贝尔实验室研究员 Ken Thompson，由于自己工作上的需求，需要一套档案存取的小型操作系统，便在 1969 年的时候以 DEC 公司（Digital Equipment Corporation）的计算机 PDP-7 为硬件基准，设计了一个适合自己工作环境的小型档案系统，其中也包含了他自行开发的一些小工具。那个系统可以说是 Multics 的最大简化版，是最早期的 UNIX 的源头。那时候还不叫 UNIX。

Ken Thompson 的这个小 OS 在贝尔实验室流行，大家觉得挺好用的。1973 年，贝尔实验室里的一个名为 Dennis Ritchie 的同事，组成一个研究小组改版这个系统，使用编译性能更高的 C 语言来重新编写系统，最后命名为 UNIX，UNIX 就这样诞生了。那个时候的 UNIX 并不是每个人都能够使用的，只有一些工程师才懂得使用。

从那时开始，很多教育机构、大型企业都投入研究 UNIX，并取得不同程度的研究成果。这样一来，软件的经济利益和版权问题来了。

UNIX 爱好者 Richard M. Stallman 先生提出 Open Source 的概念，认为大家应该共享自己的程序，这样程序会有很多人参与校验，在不同的平台进行测试，编写出更好的程序。为了自己的理想，Stallman 在 1984 年创立了 GNU 与自由软件基金会（Free Software Foundation, FSF），并且创作了许多“自由软件”来供大众使用。此外，对于其作品以自由（Free）的 GNU General Public License（GPL）的授权模式供大众使用。这个 FSF 的核心观念是“版权制度是促进社会进步的手段，版权本身不是自然权力”。Stallman 的 GNU General Public

License 一直强调 Free 这个字眼。他是这样说的：“Free software” is a matter of liberty, not price. To understand the concept, you should think of “free speech”, not “free beer”. “Free software” refers to the users freedom to run, copy, distribute, study, change, and improve the software。大意是说，自由软件并不是指“免费”的，而是指具有“自由度， freedom”的软件，什么是自由度呢？也就是你在取得这个软件之后，可以进行修改、进一步发表与复制在不同的计算机平台上。这无疑是个好消息。因为如此一来，你所拿到的软件可能原先只能在 UNIX 上面运行，但是经过修改源代码之后，你将可以拿它在 Linux 或者是 Windows 上面运行。Stallman 先生更直接地对自由度下了这样的定义：你可以根据任何你想要的提议（ purpose ）来运行这个 Free 的程序；你可以在理解了这段 Free 程序的运行之后，将它修改成你所想要的功能；你可以将自己修改过的 Free 的程序再次发布，以帮助你的朋友们；你可以将这段 Free 程序进行改良，并将改良过的程序公开发布，以造福社群。

简单说，GPL 授权具有几个特点：任何软件挂上 GPL 授权之后，即为自由的软件，任何人均可取得。同时，亦可取得其原始码（ Source Code ）；取得 GPL 授权软件后，任何人均可修改原始码，以符合自己的喜好；除此之外，经过修改的 Source Code 应回报给网络社会，供大家来参考。

GNU GPL 的出现为 Linux 诞生奠定了基础。

1.1.2 MINIX 系统

MINIX 系统是由 Andrew S.Tanenbaum (AST) 开发的。AST 在荷兰 Amsterdam 的 Vrije 大学数学与计算机科学系统工作，是 ACM 和 IEEE 的资深会员，共发表了 100 多篇文章，编写了 5 本计算机书籍。AST 虽出生在美国纽约，但他是荷兰侨民。他在纽约上的中学、在 M.I.T 上的大学、在加州大学 Berkeley 分校念的博士学位。由于读博士后的缘故，他来到了家乡荷兰。从此就与家乡一直有来往。后来就在 Vrije 大学开始教书、带研究生了。荷兰首都 Amsterdam 是个常年阴雨绵绵的城市，而对于 AST 来说，这最好不过了，因为这样他就可以待在家里摆弄他的计算机。MINIX 是他在 1987 年编写的，主要用于学生学习操作系统原理。到 1991 年时版本是 1.5。目前主要有两个版本在使用：1.5 版和 2.0 版，当时该操作系统在大学使用是免费的，但其他用途不是，当然目前都已经是免费的，可以从许多 FTP 上下载。Linux 继承了 MINIX 系统。

1.1.3 POSIX 标准

POSIX (Portable Operating System Interface for Computing Systems) 是由 IEEE 和 ISO/IEC 开发的一簇标准。该标准是基于现有的 UNIX 实践和经验，描述了操作系统的调用服务接口，用于保证编制的应用程序可以在源代码一级上在多种操作系统上移植运行。它是在 1980 年一个 UNIX 用户组 (usr/group) 的早期工作基础上取得的。该 UNIX 用户组试图将 AT&T 的系统 V 和 Berkeley CSRG 的 BSD 系统的调用接口之间的区别重新调和集成，从而于 1984 年产生了/usr/group 标准。1985 年，IEEE 操作系统技术委员会标准小组委员会 (TCOS-SS) 开始在 ANSI 的支持下责成 IEEE 标准委员会，制定有关程序源代码可移植性操作系统服务接口正式标准。到了 1986 年 4 月，IEEE 就制定出了试用标准。第一个正式标准是在 1988 年 9



月批准的（IEEE 1003.1-1988），也是以后经常提到的 POSIX.1 标准。

1989 年，POSIX 的工作被转移至 ISO/IEC 社团，并由 15 个工作组继续将其制定成 ISO 标准。到 1990 年，POSIX.1 与已经通过的 C 语言标准联合，正式批准为 IEEE 1003.1-1990 和 ISO/IEC 9945-1:1990 标准。

POSIX.1 仅规定了系统服务应用程序编程接口（API），仅概括了基本的系统服务标准。因此，期望对系统的其他功能也制定出标准。这样 IEEE POSIX 的工作就开始展开了。于 1990 年，刚开始有十个批准的计划在进行，有近 300 多人参加每季度为期一周的会议。着手的工作有命令与工具标准（POSIX.2）、测试方法标准（POSIX.3）、实时 API（POSIX.4）等。到了 1990 年上半年，已经有 25 个计划在进行，并且有 16 个工作组参与进来。与此同时，还有一些组织也在制定类似的标准，如 X/Open、AT&T、OSF 等。

在 20 世纪 90 年代初，POSIX 标准的制定正处在最后投票敲定的阶段，也就是在 1991~1993 年间。此时正是 Linux 刚刚起步的时候，这个 UNIX 标准为 Linux 提供了极为重要的信息，使得 Linux 能够在标准的指导下进行开发，能够与绝大多数 UNIX 系统兼容。在最初的 Linux 内核代码中（0.01 版、0.11 版）就已经为 Linux 与 POSIX 标准的兼容做好了准备工作。在 0.01 版的内核/include/unistd.h 文件中就已经定义了几个有关 POSIX 标准要求的常数符号，并且在注释中就写到“ok，这也许是个玩笑，但我正在着手研究它呢”。

1991 年 7 月 3 日在 comp.os.minix 上发布的 post 上就已经提到了正在搜集 POSIX 的资料。其中透露了他正在进行 Linux 系统的开发，并且在 Linux 最初的时候已经想到要实现与 POSIX（UNIX 的国际标准）的兼容问题。

1.1.4 Linux 的诞生

1981 年，IBM 公司推出享誉全球的微型计算机 IBM PC。在 1981~1991 年间，MS-DOS 操作系统一直是微型计算机上操作系统的主宰。此时，计算机硬件价格虽然逐年下降，但软件价格仍然是居高不下。当时 Apple 的 MACs 操作系统可以说是性能最好的，但是其天价没有人能够轻易靠近。

当时的另一个计算机技术阵营是 UNIX 世界。但是 UNIX 操作系统就不仅是价格贵的问题了。当时 UNIX 没有一个标准，UNIX 百家争鸣，出现了许多 UNIX 的版本。为了寻求高利率，UNIX 经销商将价格抬得极高，PC 小用户根本不能靠近它。曾经一度受到贝尔实验室的许可，可以在大学中用于教学的 UNIX 源代码一直被小心地守卫着不许公开。对于广大的 PC 用户，软件行业的大型供应商始终没有给出有效的解决办法。正在此时，出现了 MINIX 操作系统，并有一本详细的书描述它的设计实现原理。由于 AST 的书写得非常详细，并且叙述有条有理，几乎全世界的计算机爱好者都在看这本书，以理解操作系统的工作原理。其中包括 Linux 系统的创始者 Linus Benedict Torvalds。当时（1991 年），Linus Benedict Torvalds 是赫尔辛基大学计算机科学系的二年级学生，也是一个靠自学成才的 hacker。21 岁的芬兰年轻人喜欢搞计算机，测试计算机的性能及它的局限性，可惜缺乏一款专业级的操作系统。MINIX 虽然很好，但只是一个用于教学目的的简单操作系统，而不是一个强有力的实际操作系统，完全不能应用于平时的工作。

到 1991 年，GNU 计划已经开发出了许多工具软件。最受期盼的 Gnu C（Stallman 开发）

编译器已经出现，但还没有开发出免费的 GNU 操作系统。即使是 MINIX 也开始有了版权，需要购买才能得到源代码。而 GNU 的操作系统 HURD 一直在开发之中，而且并不能在几年内完成。对于 Linus 来说，已经不能等待了。从 1991 年 4 月份起，他开始酝酿并着手编制自己的操作系统。刚开始，他的目的很简单，只是为了学习 Intel 386 体系结构保护模式运行方式下的编程技术。但后来 Linux 的发展却完全改变了他的初衷。

1991 年初，Linus 开始在一台 386sx 兼容微机上学习 MINIX 操作系统。通过学习，他逐渐不能满足 MINIX 系统的现有性能，并开始酝酿开发一个新的免费操作系统。根据 Linus 在 comp.os.minix 新闻组上发布的消息，我们可以知道他逐步从学习 MINIX 系统到开发自己的 Linux 的过程。

Linus 第 1 次向 comp.os.minix 投递消息是在 1991 年 3 月 29 日。题目是“gcc on MINIX-386 doesn't optimize”，是有关 GCC 编译器在 MINIX-386 上运行的优化问题，由此可知，Linus 在 1991 年的初期，已经开始深入研究 MINIX 系统，并在这段时间有了改进 MINIX 操作系统的想法，而且在进一步学习 MINIX 系统中，逐步演变为想自己重新设计一个基于 Intel 80386 体系结构的新操作系统。他在回答有人提出 MINIX 上的一个问题时，所说的第一句话是“阅读源代码”。他认为答案就在源程序中。这也说明了对于学习系统软件来说，程序员不光需要懂得系统的基本工作原理，还需要结合实际系统，学习实际系统的实现方法。因为理论毕竟是理论，其中省略了许多枝节，而这些枝节问题虽然没有太多的理论含量，却是一个系统必要的组成部分，就像麻雀身上的一根羽毛。

从 1991 年的 4 月开始，Linus 几乎花了全部时间研究 386-MINIX 系统，并且尝试着移植 GNU 的软件到该系统上，如 GNU GCC、bash、gdb 等。并于 4 月 13 日在 comp.os.minix 上发布说自己已经成功地将 bash 移植到了 MINIX 上，而且已经爱不释手、不能离开这个 shell 软件。

1991 年 8 月 25 日，Linus 在 comp.os.minix 发布了一条消息，说向所有 MINIX 用户询问“What would you like to see in minix？”，在该消息中他首次透露出正在开发一个免费的 386（486）操作系统，并且说只是兴趣而已，代码不会很大，也不会像 GNU 的那样专业。开发免费操作系统这个想法从 4 月份就开始酝酿了，希望大家反馈一些对于 MINIX 系统中喜欢哪些特色，不喜欢什么等信息。由于实际的原因和其他一些原因，新开发的系统刚开始与 MINIX 很像，使用了 MINIX 文件系统，并且已经成功地将 bash 1.08 版和 gcc 1.40 版移植到了新开发的系统上，而且经过几个月的奋斗，就可以运行在新开发的系统上了。

最后，Linus 申明他开发的操作系统没有使用一行 MINIX 的源代码；而且由于使用了 386 的任务切换特性，所以该操作系统不好移植，其实是不可以移植，并且只能使用 AT 硬盘。当时，Linus 并没有考虑过 Linux 的移植性问题，但是目前 Linux 几乎可以运行在任何一种硬件体系结构上。

到 1991 年的 10 月 5 日，Linus 在 comp.os.minix 新闻组上发布消息，正式向外宣布 Linux 内核系统的诞生（Free minix-like kernel sources for 386-AT）。这段消息可以称为“Linux 的诞生宣言”，并且一直广为流传。因此，10 月 5 日对 Linux 社区来说是一个特殊的日子，许多后来 Linux 的新版本发布时都选择了这个日子。



1.1.5 Linux 的命名由来和发音

Linux 操作系统开始的时候并不叫 Linux。在 Linus 需要把他新开发的操作系统发布到 Ftp 上的时候，他把它命名为 FREAX，其英文含义是怪诞的、怪物、异想天开等意思。可是 ftp.funet.fi 服务器的管理员 Ari Lemke 很不喜欢这个名称。他认为既然是 Linus 的操作系统就取其谐音 Linux 作为该操作系统的目录，于是 Linux 这个名称就开始流传下来了。

目前，很多不了解 Linux 的人，都有一个误区，把 Linux 读为 “/linju:ks/”，真正的读音应该为 “/li:nəks/”。就因为一个发音，Linus 录下了一段自己对 Linux 的发音，以矫正世界各地的发音，这段音频可以在 Google 上搜索得到。

1.1.6 Linux 的标志

相信刚开始接触 Linux 的读者第一眼看到的应该就是那个可爱的小企鹅。也许你们会问怎么会使用企鹅作为 logo。选择这只名为 Tux 的企鹅作为 Linux 的标志，是有一段故事的。

在 1996 年，5 岁的 Linux 操作系统开始在世界范围内被广泛应用，Linus 开始考虑给这个操作系统创作一个 logo。他的思绪跑到动物世界去寻找：信天翁、鸭嘴兽、老鹰，还曾经想到过狐狸。就这样，一只只动物在 Linus 脑中闪过。最终，他的灵感闪光，他想起曾经在澳大利亚被一种鸟咬过，咬他的不是别的鸟，就是企鹅。

选定了企鹅作为 logo 之后，Linus 在 1996 年的 Linux 内核邮件列表中讨论寻找一只丰满敦实的企鹅时形容道：“不要太胖，而且应该是坐着的姿态，想象一下装满豆子的袋子。”当时 22 岁的电子工程本科生 Larry Ewing 在学校科学计算研究所工作时看到了邮件列表里关于 logo 的讨论。这位热心的艺术家发现执笔画企鹅比做毕业论文有趣。

Ewing 使用 GIMP 创建了企鹅的基本形态，准备公布的时候，他突然想到，这只企鹅如果出在别人手里会是什么样？于是在保留 Ewing 和 GIMP 名字的前提下，他允许用户自由添加、修改这只企鹅的造型，也衍生了现在诸多企鹅的造型。目前，Ewing 是 Ximian 的开发人员，Ximian 的猴子图案也出自他笔下。

至于企鹅的名字：Tux。因为企鹅总是黑白一身，当时外界认为这个名字来自燕尾服 (Txuedo) 一词。事实上 Tux 是 “Torvalds' UNIX” 的缩写。

现在，企鹅 Tux 已经是 Linux 的标志，或者说品牌。笑容可掬的 Tux 吸引了世界众多爱好自由的电脑爱好者。Tux 自身的含义也随着 Linux 的壮大不断发展，按 Ewing 的话来讲：“这真是一件有意思的事情，我只花了 20 多个小时画了只企鹅，同时我也花了四五年时间写程序代码。但是，我所了解的企鹅远远比任何代码的含义还要深刻。”

1.2 Linux 的内核和发行



Linux 的内核的版本号指的是在 Linus 领导下的开发小组，开发出的系统内核的版本号。一般版本号为 XX.XX.XX，第一位的 XX 表示主版本号，第二位的 XX 表示次版本号，第三

位的 XX 表示所做的小修改。第二位 XX 数字，若为偶数，表示此版本为稳定版本，如 2.6.XX；若为奇数，则为测试版本，添加了一些新的功能，如 2.5.XX。

只有 Linux 内核是难以在 PC 上使用的。为了方便普通用户使用，很多大型的公司都在 Linux 的内核基础上开发了自己的操作系统。因此，Linux 操作系统的发行版本可以说是百花齐放。

1.2.1 内核的发展

1991 年 10 月 5 日，Linus 发布的第一个 Linux 版本为 0.02，当时这一版本，只有两个在终端上打印 AAAA... 和 BBBB... 的进程。这就是最初的版本。

1991 年 11 月，网络上开始有些 hacker 参与到项目中来，很快推出了 0.10 版本，接着在 1991 年 12 月推出了 0.11 版本。当时的 Linux 性能非常接近于一种可靠的、稳定的系统，因此在发布 0.13 版本的时候，Linus 把它称为 0.95 版本，同时也是告诉大家，离 Linux 1.0 版本已经不远了。

到了 1992 年，大约有 1000 人在使用 Linux，值得一提的是，他们基本上都是属于真正意义上的 hacker。

1993 年，大约有 100 余名程序员参与了 Linux 内核代码编写/修改工作，其中核心组由 5 人组成。此时，Linux 0.99 版本的代码大约有 10 万行，用户大约有 10 万左右。

1994 年 3 月 14 日，发布了 Linux 1.0 版本——正式的 Linux 内核版本。从那时候开始，Linux 内核的开发小组已成立，内核的开发和规范都是由 Linux 社区负责。由 Linus 负责领导的 Linux 内核开发小组，每隔一段时间他们都会发布内核的新版本和修订版。

1995 年 3 月，Linux 1.2.0 版本发布。此时的 Linux 可在 Intel、Digital 以及 Sun SPARC 处理器上运行，用户量也超过了 50 万，相关介绍 Linux 的 Linux Journal 杂志也发行了超过 10 万册之多。

1996 年 6 月，Linux 2.0 内核发布，此内核有大约 40 万行代码，并可以支持多个处理器。此时的 Linux 已经进入了实用阶段，全球大约有 350 万人使用。

1999 年 1 月 25 日，Linux 2.2.0 内核发布，这一内核版本添加了许多新的东西，性能得到了很大的提高，可以移植到多样的硬件上。

2001 年期间，有许多的大公司都投入到 Linux 领域，加速了 Linux 的发展。SGI 将它的 XFS 文件系统移植到 Linux 上。IBM 提供 Linux 服务支持，Linux 应用到服务器上。2001 年 1 月，Linux 2.4.0 内核发布。内核添加了数据库的支持。2002 年 Linux 2.5.0 支持 64 位机。2003 年 12 月，Linux 2.6.0 内核发布。我们目前使用的版本，基本上都是在 2.6 内核的基础上开发出来的。

1.2.2 主流发行版本

90 年代初期 Linux 开始出现的时候，仅仅是以源代码形式出现，用户需要在其他操作系统下进行编译才能使用。随着计算机软硬件的发展，计算机用户越来越多，为了方便使用，后来出现了一些正式版本，可安装不需编译，用户可以自行安装到特定的计算机上，拥有较好的操作界面。以下将简述目前比较流行的 Linux 发行版本。



1. Slackware

Slackware Linux 是由 Patrick Volkerding 开发的 GNU/Linux 发行版。与很多其他的发行版不同，它坚持 KISS(Keep It Simple Stupid)的原则，保持了 BSD 规范，就是说它有任何配置系统的图形界面工具。对于热爱计算机的人，Slackware 很具吸引力。在它的社区里边流行着这么一句：“When you know Slackware you know Linux, when you know Red Hat, all you know is Red Hat.” 现在 Slackware 的发布版本为 Slackware 12.0。

2. Red Hat Linux

Red Hat Linux 是商业上运作最成功的一个 Linux 发行套件，普及程度很高，由 Red Hat 公司发行。Red Hat Linux 中的 RPM 软件包格式可以说是 Linux 社区的一个事实标准，被广泛使用于其他 Linux 发行套件中。自从 Red Hat 9.0 版本发布后，Red Hat 公司就不再开发桌面版的 Linux 发行套件，而将全部力量集中在服务器版的开发上，也就是 Red Hat Enterprise Linux 版。2004 年 4 月 30 日，Red Hat 公司正式停止对 Red Hat 9.0 版本的支援，标志着 Red Hat Linux 的正式完结。原来的桌面版 Red Hat Linux 发行套件则与来自民间的 Fedora 计划合并，成为 Fedora Core 发行版本。Fedora 每六个月发布一个新的版本，目前的最新版本为 2008 年 11 月 25 日发布的 Fedora 10.0。

3. Debian

Debian 是一种自由操作系统，全称 Debian GNU/Linux，采用 Linux 作为内核，由 Debian 计划（Debian Project）组织维护。Debian 是一个纯粹由自由软件所组合而成的作业环境，而其背后并没有任何的营利组织支持，它的开发团队全部都是来自世界各地的志愿者，官方开发者的总数就将近 1000 名，而非官方的开发者亦为数众多。

Debian 以其坚守 UNIX 和自由软件的精神，以及其给予用户的众多选择而闻名，现时 Debian 包括了超过 18,000 个软件包并支援 11 个计算机系统结构。有很多 Linux 发行版本都继承了 Debian 系统，如 Ubuntu、Knoppix 和 Linspire 及 Xandros。

4. SUSE

SUSE（发音 /'su:sə/）是 Linux 操作系统中的一个发行版，也是德国的一个发行版，是在欧洲大陆最流行的版本之一。SUSE Linux 原是以 Slackware Linux 为基础，并提供完整德文使用界面的产品。到了 2005 年，SUSE 有三种发行版本：“OSS 版”、“试用版”及盒装零售版。其中 OSS 版是完全开放源代码的发行版本。目前，SUSE 发行版本为 2008 年 6 月 19 日发布的 SUSE Linux 11.0。

5. Ubuntu

Ubuntu（国际音标：/u: 'bu:ntu:/）是一个以桌面应用为主的 Linux 操作系统，其名称来自非洲南部祖鲁语或豪萨语的“ubuntu”一词（译为吾帮托或乌班图），意思是“人性”、“我的存在是因为大家的存在”。Ubuntu 建基于 Debian 发行版和 GNOME 桌面环境，与 Debian 的不同在于它每 6 个月会发布一个新版本。普通的桌面应用版可以获得 18 个月的支持，标为 LTS 的桌面应用版可以获得更长时间的支持。目前最新的版本为 Ubuntu 9.04 版。

它在中国也取得了可喜的发展。从 1999 年 3~4 月开始，国内涌现出不少颇具实力的中文 Linux 发行商。如中科红旗公司推出的第一个桌面应用版红旗 Linux 2.0、采用了多项新技术的蓝点 Linux 2.0、完全符合国际化标准的中文 Linux 发行版、TurboLinux 简体中文版 6.0、

Xteam Linux 3.2 等。Xteam 公司和中科红旗软件公司近期还都宣称将利用 Linux 2.4 内核推出自己的新产品。

1.2.3 GNU 通用公共许可证

GNU 计划，又称“革奴计划”，是由 Richard Stallman 在 1983 年 9 月 27 日公开发起的。它的目标是创建一套完全自由的操作系统。Richard Stallman 最早是在 net.UNIX-wizards 新闻组上公布该消息，并附带一份《GNU 宣言》等解释为何发起该计划的文章，其中一个理由就是要“重现当年软件界合作互助的团结精神”。

GNU 是“GNU's Not UNIX”的递归缩写，所表达的意思是开发出一套与 UNIX 相似而不是 UNIX 的系统。在读音方面，为避免与 gnu（非洲牛羚，发音与“new”相同）这个单词混淆，Stallman 宣布 GNU 应当发音为“Guh-NOO”(/'gnu:/)，与“canoe”发音相同，以矫正世界各地对 GNU 的发音。

UNIX 是一种广泛使用的商业操作系统名称。由于 GNU 将要实现 UNIX 系统的接口标准，所以 GNU 计划可以分别开发不同的操作系统部件。GNU 计划采用了部分当时已经可自由使用的软件，例如 TeX 排版系统和 XWindows 视窗系统等。不过 GNU 计划也开发了大批其他的自由软件，这些软件也被移植到其他操作系统平台上，例如 Microsoft Windows、BSD 家族、Solaris 及 MacOS。这都为 Linux 的诞生提供了环境。

为保证 GNU 软件可以自由地“使用、复制、修改和发布”，所有 GNU 软件都包含一份在禁止其他人添加任何限制的情况下，授权所有权利给任何人的协议条款，GNU 通用公共许可证（GNU General Public License, GPL）。这个就是被称为“反版权”（或称著佐权 Copyleft）的概念。GNU 也针对不同场合，提供 GNU 宽通用公共许可证（GNU Lesser General Public License, LGPL）与 GNU 自由文档许可证（GNU Free Documentation License, GFDL）这两种协议条款。

1985 年，为了向 GNU 计划提供技术和经济上的援助，Stallman 创建了自由软件基金会（Free Software Foundation）。

到 1990 年，GNU 计划开发出不少功能强大的软件，如 Emacs、gcc。这些工具后来都应用到了 Linux 系统。

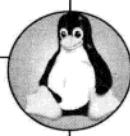
1991 年，Linus 按照 GPL 条款发布了 Linux，这使得 Linux 很快就获得了专业人士的加入，促进了 Linux 的快速发展。

GNU GPL 的精神就是开放、自由，为优秀的程序员提供展现自己才能的平台，也使他们能够编写出自由的、高质量、易理解的软件。

1.3 小结

Linux 是新生的操作系统，具有顽强的生命力。在它的发展过程中，有不少能人志士致力于 Linux 领域，促进它的发展。使用 Linux，你可以免费获得一个适合自己的操作系统，你不必担心使用它会牵涉到版权问题。





本章是 Linux 的基础知识。本章的内容包括：UNIX 的历史，Linux 的历史，Linux 发展过程中融合的一些标准如 POSIX 标准，还有作为 Linux 的指导思想的 GNU GPL 等一系列许可证。

第 2 章将简单介绍 Ubuntu Linux，其中还要介绍它的基版本 Debian，最后会说说最新发布的 Ubuntu 9.04 的特点。

1.4 习题



1. UNIX 之父是谁？Linux 之父是谁？
2. 目前流行的 Linux 有哪些？各有什么特点？
3. GNU 计划有什么作用？

Linux

第2章

Ubuntu Linux 概述

在上一章中，我们了解了 Linux 的简史。Linux 由于可以自由发行，遵守 GPL 规范，许多公司和民间组织很容易就可以获得 Linux 内核的源代码，并可以在源码的基础上开发出各种各样的 Linux 版本。Ubuntu 的出现，也是得益于 GPL，它继承了 Debian 的所有优点。接下来，我们从 Ubuntu 的简史、发行，以及它的特点方面展开简述 Ubuntu Linux。





2.1 Ubuntu 简史



Ubuntu 基于 Debian 发行版和 GNOME 桌面环境，与 Debian 的不同在于它每 6 个月会发布一个新版本。

2.1.1 Debian 的历史

Debian 是 Ubuntu 的一个父版本，Ubuntu 的发展离不开 Debian。本节从 Debian 的发展、思想和优缺点来讲述 Debian 的历史。

1. Debian 发展

Debian 于 1993 年 8 月 16 日由一名美国普渡大学学生 Ian Murdock 首次发布。在定义文件 *Debian Manifesto* 中，Ian Murdock 本着 Linux 及 GNU 的精神发行一套 GNU/Linux 发行版，并以他女友（现在为他妻子）Debra 和 Ian Murdock 自己的名字合并命名为 Debian。Debian 的读音为：/dəbi : jən/。

1994 年 1 月，也就是首个发行版本诞生 5 个月之后，官方第一次发行的 Debian 版本中出现的二进制包系统被确定为 Debian 0.91。

1.x 版本则在 1996 年发布。1996 年，Bruce Perens 接替了 Ian Murdock 成为 Debian 计划的领导者。同年，一个开发者 Ean Schuessler 提议 Debian 应在其计划与使用者之间建立一份社会契约。经过讨论，Bruce Perens 发表了 Debian 社会契约（*Debian Social Contract*）及 Debian 自由软件指引（*Debian Free Software Guidelines*），定义了开发 Debian 的基本承诺。这样，Debian 走上了快速发展的轨道。

1998 年 3 月，由于在工程怎样发展上的分歧，Bruce Perens 彻底辞职不干。1998 年 6 月份，Debian 2.0 进入测试阶段。APT 和第一个非 Linux 接口——Debian GNU/Hurd 的开发也展开了。7 月，Debian 2.0 发布，而且容量极大，达到 1500 个包共两张光盘。10 月份，Debian 考虑到 Qt 许可协议问题，决定将 KDE 从他们的发行版中剔除。

1999 年 3 月 9 日，比计划推迟了 7 天，Debian 2.1 最终发布，代号为 Slink。

2000 年后半年，Debian 对数据库和发布的管理作出了重大的改变，它重组了收集软件的过程，并创造了“测试”（*testing*）版本作为较稳定的对下一个发布的演示。同年，Debian 的开发者开始举办名为 Debconf 的年会，为其开发者和技术用家提供讲座和工作坊。8 月 15 日 Debian 2.2 发布，代号为 Potato。Debian GNU/Linux 2.2 架构于最新的稳定版 Linux 内核（2.2.16）之上，并补加了 Alan Cox 准备放入 2.2.17 版的补丁（*patches*）。2.2 版内核在可用性和稳定性方面有重大改善，也支持更多新旧的硬件产品，其的中文环境易用性大幅度提高。

2002 年 7 月 19 日，Debian GNU/Linux 3.0 发布，代号为 Woody。Woody 是 Potato 的继承者。修改了 Potato 的漏洞，支持多达 11 种的桌面环境平台，包括 GNOME 和 KDE，支持 64 位机，支持多种处理器体系结构。

2005 年 6 月 6 日，Debian GNU/Linux 3.1 发布，代号 Sarge。从此版本开始，Debian 提

供了众多安装方式，包括 U 盘安装，网络安装。

2007 年 4 月 8 日 Debian GNU/Linux 4.0 发布，代号 Etch。Debian 能够在小至电子手账、大至超级计算机的机器上运行，且运行更加流畅。

2009 年 2 月 14 日，Debian GNU/Linux 5.0 发布，代号 Lenny。这次发行中，5.0 版本加入了对 Marvell 的 Orion 平台的支持，它被用于许多存储设备，包括了新的 ARM EABI 移植，对过期的大部分软件进行了升级处理，集成 X.Org 7.3，X server 可以对绝大多数硬件进行自动配置，在安装方面也做了改动。

2. Debian 的思想

Debian 是一个纯粹由自由软件所组合而成的作业环境，而其背后并没有任何的营利组织支持，它的开发团队全部都是来自世界各地的志愿者，官方开发者的总数将近 1000 名，而非官方的开发者亦为数众多。

Debian 坚守 UNIX 和自由软件的精神，其全称为 Debian GNU/Linux。Debian 指定了一个社会公约：

- (1) Debian 将永远是自由软件。
- (2) Debian 将和自由软件社区共享成果。
- (3) 公开 Bug 报告结果我们将始终把我们整个的错误报告数据库开放给公众阅读。
- (4) 我们将优先考虑我们的用户及自由软件。
- (5) 非自由软件指导方针。

3. Debian 的优缺点

Debian 永远是自由软件，可以在网上免费获得。Debian 具有以下优点：

- (1) Debian 是极为精简的 Linux 发行版，有着干净的操作环境。
 - (2) 安装步骤简易，大部分情况下只要按回车键并一直按下去便可以顺利安装。
 - (3) 拥有方便的套件管理程序，可以让使用者容易寻找、安装、移除、更新程序，或系统升级。
 - (4) 健全的软件管理制度，包括了 Bug 汇报、套件维护人等制度，让 Debian 所收集的软件品质在其他的 Linux 发行套件之上。
 - (5) 拥有庞大的套件库，令使用者只需通过它自身所带的软件管理系统便可下载并安装套件，不必再在网路上寻找。
 - (6) 套件库分类清楚，使用者可以明确地选择安装自由软件、半自由软件或闭源软件。
- 拥有优点的同时，Debian GNU/Linux 也具有一定的缺点和不足，如下：
- 软件不能及时获得更新。
 - 一些非自由软件不能得到很好的支持。
 - Debian 的发行周期长。

2.1.2 Ubuntu 的发展

Ubuntu Linux 是 Debian 的一个改版，它继承了 Debian 的优点，使用很多在 Debian 下经过测试的优秀自由软件。Ubuntu 音标为：/u:/bu:/ntu:/，是一个以桌面应用为主的 Linux 操作系统，其名称来自非洲南部祖鲁语或豪萨语的“ubuntu”一词（译为吾帮托或乌班图），



意思是“人性”、“我的存在是因为大家的存在”。

Ubuntu 由马克·舍特尔沃斯创立，其首个版本 4.10 发布于 2004 年 10 月 20 日，代号 Warty Warthog，它以 Debian 为开发蓝本。在发布文件中声明，用户可以免费获得和使用 Ubuntu，罗列出了 Ubuntu 中使用的自由软件。

2005 年 4 月 8 日，发布 Ubuntu 5.04，代号为 Hoary Hedgehog（白发的刺猬）。

2005 年 10 月 13 日，发布 Ubuntu 5.10，代号为 Breezy Badger（活泼的獾）。

2006 年 6 月 1 日，发布 Ubuntu 6.06，代号为 Dapper Drake（整洁的公鸭）。

2006 年 10 月 26 日，发布 Ubuntu 6.10，代号为 Edgy Eft（尖利的小蜥蜴）。

2007 年 4 月 19 日，发布 Ubuntu 7.04，代号为 Feisty Fawn（烦躁不安的小鹿）。

2007 年 10 月 18 日，发布 Ubuntu 7.10，代号为 Gutsy Gibbon（胆大的长臂猿）。

2008 年 4 月 24 日，发布 Ubuntu 8.04，代号为 Hardy Heron（坚强的苍鹭）。

2008 年 10 月 30 日，发布 Ubuntu 8.10，代号为 Intrepid Ibex（无畏的山羊）。

2009 年 4 月 23 日，发布 Ubuntu 9.04，代号为 Jaunty Jackalope（得意扬扬的怀俄明野兔）。

预定在 2009 年 10 月发布 Ubuntu 9.10，代号为 Karmic Koala（幸运的无尾熊）。

细心的读者都会发现，每个版本的发布时间相差不会太远，一般都是半年，而且发布的版本号，都是由发布的年份和月份组成的。还会发现代号的命名都是很有意思的，代号都是由两个单词组成的，而且两个单词的第一个字母都是相同的，在命名上，有这样的一个规定，第一个单词为形容词，第二个单词为动物的名词，这点和 Debian 的代号命名有所不同，Debian 的开发代号来源于电影《玩具总动员》。

2.2 Ubuntu 的特点

Ubuntu 的开发目的是为了使个人电脑变得简单易用，同时也提供针对企业应用的服务器版本。Ubuntu 的每个新版本均会包含当时最新的 GNOME 桌面环境，通常在 GNOME 发布新版本后一个月内发行。与其他基于 Debian 的 Linux 发行版，如 MEPIS、Xandros、Linspire、Progeny 和 Libranet 等相比，Ubuntu 更接近 Debian 的发展理念，它主要使用自由、开源的软件，而其他发行版往往会附带很多闭源的软件。

Ubuntu 拥有庞大的社区群支持它的开发，用户可以及时获得技术支持，软件更新快，系统运行稳定，发布的普通应用版可以获得长达 18 个月的技术支持，而普通桌面版可以获得 3 年的技术，而服务器版的可以获得更长技术支持的时间为 5 年。

Ubuntu 具有优秀的软件管理软件 Synaptic，方便更新、安装、删除软件。安装“傻瓜化”，使用 Sudo 操作防止用户的错误操作。

2.3 Ubuntu 的发行

Ubuntu 每半年发行一个新的版本，版本号由发布年月组成。Ubuntu 会发行长期支援版本

(LTS)，更新维护的时间比较长，大约 2 年会推出一个正式的大改版版本。LTS 面向企业用户，有别一般版本的 6 个月支援。代号为“Dapper Drake”的 Ubuntu 6.06 LTS 是第一个获得长期支持的版本，Canonical 公司计划对 6.06 的桌面系列版本提供 3 年的更新及付费技术支持服务，对服务器版则提供 5 年的支持。

Ubuntu 的支持者众多，而且 Ubuntu 遵循着自由软件的精神，因而出现了比较多的衍生版本。目前 Ubuntu 正式支援的衍生版本包括如下。

(1) **Kubuntu**: 使用和 Ubuntu 一样的软件库，但不采用 GNOME，而是采用 KDE 作为默认的桌面环境，以满足偏爱 KDE 的 Ubuntu 用户。

(2) **Edubuntu**: 是为教育量身定做的发行版，包含很多教育软件，可以帮助教师方便地搭建网络学习环境，管理电子教室。

(3) **Xubuntu**: 属于轻量级的发行版，使用 Xfce4 作为默认桌面环境，与 Ubuntu 采用一样的软件库。

(4) **Ubuntu Server Edition**: 自 Ubuntu 5.10 版 (Breezy Badger) 起与桌面版同步发行，可用做多种服务器，如电邮服务器、基于 LAMP 的网站服务器、DNS 服务器、文件服务器与数据库服务器等。与桌面版本相比，服务器版的光盘映像较小，运行时对硬件要求较低，最少需要 500MB 硬盘空间和 64MB 内存。服务器版通常不提供任何桌面环境。

(5) **Ubuntu Studio**: 适合于音频、视频和图像设计的版本。

(6) **Ubuntu JeOS**: 一个高度精简的、专门面向虚拟化应用的版本。

Ubuntu 非正式的衍生版本包括：

(1) **nUbuntu**: 专注于安全工具的版本。

(2) **Ubuntu Lite**: 为旧电脑而设的版本。

(3) **zUbuntu**: 为 IBM zSeries 主机移植的版本。

(4) **Ebuntu**: 基于 Enlightenment 0.17 桌面环境并附有视窗管理员的 Ubuntu 修改版。

(5) **Fluxbuntu**: 基于 Fluxbox 桌面环境的修改版。

(6) **Gnoppix**: 基于 Ubuntu Live CD 而研制的以 GNOME 为默认桌面环境的 Live CD。

(7) **Dubuntu**: 由中国爱好者开发的 Ubuntu 改进版，能够更好地支持中文，并且添加了很多软件的 Live CD。

(8) **hiweed**: 由中国爱好者开发的 Ubuntu 定制版本，根据需要对软件包进行了适当的增删。

(9) **gOS**: 基于 Enlightenment 桌面环境，整合 Google 多数的线上服务的版本（并非由 Google 官方所开发）。

(10) **Linux Mint 5 Elyssa**: 基于 Ubuntu 8.04 的 Linux 发行版，目标是提供一种更完整的即刻可用体验。

(11) **Ubuntu Eee (eeeXubuntu)**: 专为华硕的 Eee PC 定制。

读者可以根据自己不同的爱好选择发行版。



2.4 Ubuntu 9.04 新功能

Ubuntu 9.04 使用了最先进的 GNOME 2.26（或 KDE 4.2、或 Xfce 4.6）桌面环境，3 维效果使用了 Compiz Fusion 0.8.2 的最新版本。Ubuntu 9.04 版本使用了 EXT4 新文件系统（虽然目前还不是默认配置），明显加快了启动速度（只需几十秒钟），采用了 OpenOffice.org 3.0.1 办公套件，极大地改善了 Ubuntu 9.04 版本的办公性能。由于该版本基于 Linux 2.6.28 内核，硬件驱动性能大为改善。该版本的安装过程也有许多新特点，如表 2-1 所示。

表 2-1 Ubuntu 9.04 的新特点

Ubuntu 版本	在 Ubuntu 9.04 中，系统更新了以下软件的版本：OpenOffice 3.0、最新的 Skype、最新的 Adobe Flash、GNOME 2.26、Brasero 2.26（CD 烧录软件）、X.Org server 1.6（支持多种视频卡，如 ATI）。具体的改进包括 EXA 加速默认的，2D 支持 R6/R7 系列，3D 支持 R5 系列） 系统性能得到很大的提高，如更快的启动时间（测试报告上说，少于 25 秒）、更好地对显示器支持、更统一的通知和系统信息和支持 Ext4 文件系统
Netbook Remix 版本	此版本拥有更快的启动时间、加强了电源管理、更容易的网络切换、更直观的图标，以及其他设计变更的功能。其天生支持 Acer Aspire One, Asus Eee PC 1000, Dell Mini 9
Ubuntu 服务器版本	此版本使网络、打印、文件、数据库、邮件服务更加高效。Ubuntu 9.04 服务器版本集成了来自 Ubuntu 社区最近的、最稳定的开放源代码应用软件，并打包供给用户下载，使用和重新发布。新增强的功能包括：使用 KVM 改进了可视化；Samba 文件服务器提供聚类支持，使用集成软件包 Dovecot—Postfix 的邮件服务器使得发送邮件更加简单 Ubuntu 9.04 服务器版预演了 Ubuntu 企业与云计算。Ubuntu 是第一个支持商业分布式的系统，使企业能够在防火墙内建立云计算环境。通过 Ubuntu 9.04 服务器版本，企业可以能够在不需要考虑到数据和把数据移动到外部的云计算提供商的安全问题，而探索云计算的利益。经过去年的成功测试，Ubuntu 9.04 服务器版本支持亚马逊弹性计算云（Amazon Elastic Compute Cloud，简称 EC2）。

除以上亮点之外，Ubuntu 9.04 开始支持 ARM 和低端的处理器，并支持手提电脑的处理器体系结构，如 ARMv5EL 和 ARMv6EL-VFP。



2.5 小结

Ubuntu 是基于 Debian GNU/Linux 而开发出来的发行版本，Ubuntu 的初衷是作为 Debian 的一个测试平台，向 Debian 提供通过测试的稳定软件，并且希望 Ubuntu 中的软件可以很好地与 Debian 兼容。由于它的易用性，而且获得众多社区的支持，Ubuntu 发展成了一款不错且流行的 Linux 发行版本。

本章主要介绍了 Debian 和 Ubuntu 的历史。Ubuntu Linux 每半年发行一个新的版本，并且提供长期支持。Ubuntu Linux 为用户提供了一个简单方便的操作平台，并宣布将永远免费。Ubuntu 9.04 的发布带给了用户一个全新的感受。

第3章将讲述安装Ubuntu Linux需要准备的工作，包括了解自己的硬件，规划自己的硬盘分区，选择一种适合自己的安装方式，了解安装系统时需要注意的问题。

2.6 习题



1. Ubuntu 怎么命名版本号？它的代号有什么规定？
2. Ubuntu 有哪些特点？
3. Ubuntu 有哪些正式的发行版本？

Linux

第 2 部分 Linux 的安装

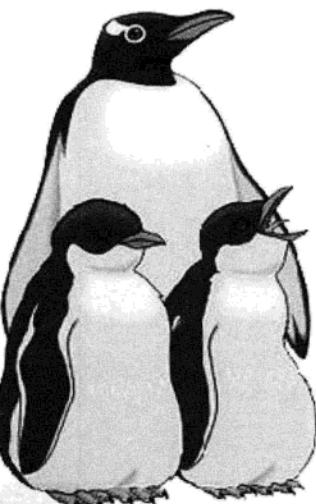
第 3 章

安装前的准备

本章主要介绍 Ubuntu 对硬件和软件的要求，帮助读者选择适合自己的 Ubuntu Linux 系统，规划磁盘分区，规划安装目录。

安装系统，首先需要了解自己的需求，了解操作系统的要求，了解自己已经拥有的设备。针对不同的用户，有着不同的需求；而系统也有它的要求，你不能使用一块几十年前的主板来运行现在的 Windows，你也不能使用一张已经停产的声卡能让你的计算机播放出美妙的音乐。通过这样的介绍之后，相信读者能够很快就掌握安装 Ubuntu 的技巧。

在 Ubuntu 的安装中，其中最重要的一点就是处理好硬盘的分区问题，我们将重点讲述硬盘分区的策略。





3.1 硬件需求

随着计算机硬件的快速发展，软件不断升级，你需要了解自己的硬件，才能在安装 Ubuntu 的过程中得心应手。尽管说安装 Linux 的最低配置是 500M 的硬盘，4M 的内存，80386SX 的处理器，相信很少有人使用这样的配置。这样的最低配置可以毫无阻碍地运行 Linux 的字符界面，要是运行桌面版的 Linux，那是不可行的。有条件我们也不可虐待自己，让计算机“跑”快点，工作更加有效率。目前，官方已在 2009 年 4 月 23 日发布了 Ubuntu 9.04，这个版本的桌面版支持炫目的 3D 效果桌面，它的最低配置为 700MHz 处理器和 256MB 内存。在最低配置上运行 Ubuntu 9.04，系统会很慢。从各方面考虑，这章将教读者怎么选择适合自己的系统，并安装 Ubuntu 系统。

因此，安装之前我们需要先考虑所需的计算机类型。Linux 可以安装在各种硬件上，如 ARM 处理器；Motorola 680X0 处理器；8086 处理器；DEC Alpha 芯片；Acorn 计算机；Power Mac 和基于 Intel 的 PC。我们不是为了安装 Ubuntu 才去买这些计算机，而是在安装 Ubuntu 的时候，我们可以很顺利地进行，并能在安装之后，可以享受到 Ubuntu 带给我们的快感。

以下是 Ubuntu 安装的系统要求：

- 700 MHz 或更好的处理器。
- 3GB of available disk space 达 3GB 的可用磁盘空间。
- 256MB of memory (RAM) 256MB 的内存 (RAM)。
- CD-ROM drive CD - ROM 驱动器。
- Ethernet interface 以太网接口。
- VGA graphics interface VGA 图形界面。

如果你觉得这样的系统配置很高，你可以选择安装 Xubuntu Linux 系统，它的要求如下。

- 500 MHz 或更好的处理器。
- 1.5GB of available disk space: 1.5GB 的可用磁盘空间。
- 192MB of memory (RAM) for installation but only 128 MB of RAM to run from the installation CD: 192MB 的内存 (RAM) 的安装，但只有 128 MB 的 RAM 运行从安装 CD。
- CD-ROM drive: CD - ROM 驱动器。
- Ethernet interface: 以太网接口。
- VGA graphics interface: VGA 图形界面。

Xubuntu Linux 安装的时候需要 192MB 的内存，可是一旦你安装成功，Xubuntu 运行时只需 64MB 的内存。

以上所说的是最低配置。计算机配置也不是越高越好，安装 Windows 有经验的人都知道，我们的硬件要运行良好，还需要有驱动程序 (Driver)。Linux 的发展还算快，而且 Linux 是由计算机高手一起开发维护的，有些新的硬件在 Linux 下还没有相应的驱动程序。那样的话，我们就不能使用这些新硬件。记得 2005 年的时候，有个朋友为了安装 FC3，费了很大的劲刻录了 5 张 CD，打算安装在他的计算机上，可是后来发现他自己的硬盘是 SATA 接口的，当



时 FC3 还没有这样的一个驱动。这给我们一个教训，安装系统之前，一定要考虑好，不要做无用功。

以下是笔者对安装 Ubuntu 的硬件需求的了解，在这里做个简要说明。

3.1.1 CPU

Ubuntu 9.04 支持几乎所有用在个人电脑上的基于 x86 的处理器，包括所有的 Intel 的 Pentium 系列，还包括 32 位的 AMD 和 VIA（前身为 Cyrix）处理器，以及 Athlon XP 和 Intel P4Xeon 多处理器。当然支持 i486 和之后发展的处理器。你的处理器是否为以下其中一种体系结构：Intel x86, AMD64 & Intel EM64T, IBM/Motorola PowerPC, Sun SPARC, HP PA-RISC, Intel IA-64。你的处理器属于上述的其中一种，那就有安装 Ubuntu 的基本条件了。

由于考虑到目前的移动设备的快速发展，Ubuntu 9.04 添加了对 ARM 体系结构的支持。Ubuntu 9.04 的安装支持在基于 ARM 的 MID 设备和运行在 ARMv5EL 或 ARMv6EL-VFP 体系结构的低端笔记本上安装。

多处理器体系结构 (Multiprocessor support，又称为“symmetric multiprocessing”或 SMP) 在 Ubuntu 9.04 上也得到了很好的支持。

3.1.2 硬盘

Ubuntu 9.04 对硬盘的要求不是很高。只要 500MB 的硬盘就可以了。目前 Ubuntu 9.04 都支持 IDE/ATA 接口、SATA 接口，以及 SCSI 接口的硬盘。目前，在 PC 上都是这几种硬盘，因此硬盘方面，相信读者已具备 Ubuntu 对硬盘的要求。对于多媒体应用的计算机，最低的配置为 1G 硬盘安装文本界面，5G 安装图形界面。

其实，Ubuntu 的推荐配置，主要是考虑到能够稳定运行 Debian GNU/Linux 的系统配置。值得注意的是，分区/var 保存除了经常使用到的文件如日志文件，还保存了 Ubuntu 特定的状态信息。在安装时，`dpkg` 包含了所有的安装包的信息很容易就耗费了 40MB 的磁盘空间，而且 `apt-get` 会在安装软件包的时候，把它们存放到 /var，这样你就需要分配 200MB 的空间给它。如果安装图形界面系统的话，就需要更多的磁盘空间。

3.1.3 内存

内存方面，普通的安装只要 56MB 的内存足矣。在 s390 上安装 Ubuntu 只要少于 20MB 的内存，要是在 i386 和 amd64 上安装只要 48MB 的内存。对于多媒体应用的计算机，推荐的配置为 256MB 的内存安装文本界面，512MB 的内存安装图形工作界面。

3.1.4 显示卡

如果只是使用文本界面的话，那么对显示卡倒没有多大的要求，几乎所有类型的显卡都可以正常运行，如 ISA、VGA。要是安装的图形界面，就要有一张比较好的显示卡，对于多媒体方面，那要求就更高了。Ubuntu 9.04 的三维立体桌面也要求一张比较好的显示卡。建议你使用知名厂家的显示卡芯片组，如 ATI、Ark、S3、Nvidia TNT 或是 Trident。

要是你已经拥有一张显示卡，那你可以到网上去搜索一下，看看你的显示卡是否获得

Linux 的支持，无论显示卡的驱动是由厂家提供，还是由 Linux 社区提供，你都可以下载下来，为你安装相应的驱动做好准备。

3.1.5 光驱

从 Ubuntu 发布 Live CD 之后，光驱变得并不是那么重要了。只要你有网络，就可以从网络上安装，也可以从他人的计算机上安装你的 Ubuntu 系统，你可以选择从硬盘启动来安装 Ubuntu。有光驱的，安装就很方便。目前，一般的光驱，例如 IDE 接口、CD-R、CD-RW、DVD-ROM 或是 SCSI 光驱都可以支持 Linux。

3.1.6 网卡

多数 ISA 或是 PCI 接口的网卡都可以在 Ubuntu 上正常使用，速度为 10MB/s 或是 100MB/s。在国内比较常用的网卡是 3Com、D-Link 和 RealTek。在启动 Live CD 的时候，系统引导会自动检测你的网卡，一般来说，都可以检测到你的网卡型号，并安装相应的网卡驱动模块。要是有些网卡在引导时没有被检测到，也许是因为打开了网卡上的“即插即用”的功能，这样的话，你可以使用系统附带的通用程序（通常为 DOS 程序）来关闭 PnP 功能。

现在是网络时代，网卡一定是要准备的。Ubuntu 还支持无线网卡，上网本的网卡也同样得到了 Ubuntu 的支持。安装系统的时候，引导程序都会自动检测网卡，并在引导程序中给你推荐最佳的配置，以方便用户可以获得流畅的网络，体验网络带来的快乐。

3.1.7 键盘与鼠标

Ubuntu 支持多数类型的键盘，我们一般使用都是美国英语式的键盘布局。在安装的时候可以设置，也可以在安装之后配置。

Ubuntu 对鼠标也没有多大的要求，它支持 PS/2 接口，AT 接口，串口和 USB 接口的鼠标。建议第一次使用 Linux 的读者不要使用 USB 接口的键盘和鼠标。

3.2 软件需求



具备了硬件设备，接下来，我们就需要有软件的支持。没有软件支持的硬件，是不能够工作的，我们称之为裸机。对于安装系统而言，我们需要操作系统的映像文件或系统启动光盘。

3.2.1 软件下载

由于 Ubuntu 是免费下载的，你可以从官方的网站 www.ubuntu.com 或是到中国的官方网站 <http://www.ubuntu.cn> 下载。作为一般的桌面用户，可选择桌面版本的 Ubuntu 镜像文件下载，下载位置可以选择日本的服务器或是上海的服务器，下载速度可以达到 90KB/s 左右。

把镜像文件下载下来存放到了你的电脑里面，若你还没有电脑存放，你可以从网上购买 CD



或 DVD 发行的版本。你到任何一部安装有其他操作系统的计算机，只要能上网就可以下载到镜像文件。若读者从 Windows 下转到 Ubuntu Linux，那可以把镜像文件保存到你的 Windows 文件系统下。

3.2.2 光盘刻录

尽管 Ubuntu 提供了 Live CD 启动的功能，可以从硬盘上安装 Ubuntu，也可以从网络上安装 Ubuntu，但对一般用户来说，还是使用光盘安装比较方便。下载下来的镜像文件大小为 698MB，一张 CD 光盘足以容下它。你可以在 Mac 下，使用 Apple's Disk Utility 软件来刻录启动光盘，在 Windows 下，一般我们会使用 Nero。接下来，笔者将简述在 Windows 下使用 Nero 刻录启动光盘，在网上可以很方便地获取到它的安装程序。把 Nero 下载到你的 Windows 下，并安装它。启动 Nero 进入主界面如图 3-1 所示。

在界面上端的标签中，选择“翻录和刻录”命令，可以看到提示框中有一个“刻录数据光盘”的选项。插入空白的 CD 光盘到刻录机，选择“刻录数据光盘”选项，就可以进到如图 3-2 所示的界面。在左边的栏目框中选择“映像 项目 复制”选项，在右边的栏目框中选择“光盘映像或保存的项目”选项。选择你下载并保存在 Windows 下的 ISO 文件，加载到 Nero 中。这样就可以看到如图 3-3 所示的刻录界面了，选择刻录机的类型，磁盘类型为 CD。还需要注意选择写入的速度，Nero 自动选择的写入速度一般都比较高，需要自己根据光盘的最大写入速度来设置写入速度。菜单是需要单击左边的隐藏按钮才会弹出。设置好了写入速度，就可以单击“刻录”按钮，过五六分钟，一张 Ubuntu 启动光盘就刻录成功了。



图 3-1 Nero 主界面



图 3-2 刻录界面

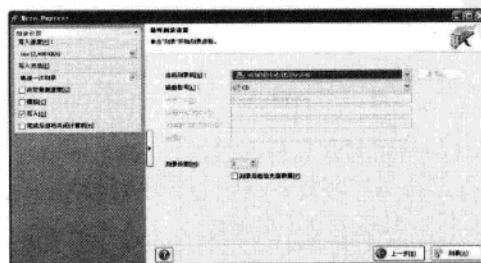


图 3-3 刻录界面

完成了 Ubuntu 9.04 启动光盘的刻录。现在软件和硬件的条件我们都有了，那我们自己的需求是什么呢？读者还需要了解一些 Linux 下的文件系统的相关知识和硬盘分区知识。

3.3 安装策略



此节将简述用户安装时需要的基础知识，了解自己的计算机配置，了解 Linux 的文件结构，了解 Linux 下的系统分区，而且重点讲述针对 Ubuntu 安装的磁盘分区方法和步骤。

3.3.1 记录硬件信息

记录硬件的信息有助于你安装，加快安装的速度，提高安装系统的成功率。表 3-1 列出了读者在安装 Ubuntu 需要记录的硬件信息。

表 3-1 硬件信息

硬盘	<ul style="list-style-type: none"> ● 硬盘空间的大小 ● 多磁盘的，磁盘的顺序情况 ● 硬盘接口是 IDE（也可以说是 PATA）或者 SATA，还是 SCSI ● 可用的空间 ● 分区情况 ● 哪个是已安装其他系统的分区 	网络卡	<ul style="list-style-type: none"> ● 制造商和型号 ● 适配器的类型
显示器	<ul style="list-style-type: none"> ● 制造商和型号 ● 所支持的分辨率 ● 垂直刷新率 ● 水平刷新率 ● 所支持的色彩深度或颜色数 ● 屏幕的大小 	打印机	<ul style="list-style-type: none"> ● 制造商和型号 ● 所支持打印的分辨率
鼠标	<ul style="list-style-type: none"> ● 类型：serial, PS/2, or USB ● 端口号 ● 制造商 ● 按键的数目 	视频卡	<ul style="list-style-type: none"> ● 制造商和型号 ● 是否拥有视频缓存 ● 所支持的分辨率和颜色深度

还有很多其他外设，针对其他外设，不同的外设有相应的特定配置要求。由于外部设备的种类和个数多种多样，在这里就不一一罗列了。

3.3.2 安装方式的选择

Ubuntu 发布的 Linux 版本，总共大小才 600 多 MB，你可以从官方网站下载，也可以通过付费购买 CD 或 DVD 获得 Ubuntu 的发布版本。从 Ubuntu 6.06 版本开始，Ubuntu 提供了 Live CD 版本的 Ubuntu，在 Ubuntu 9.04 中，Live CD 版本已包括在发布版本中。

你可以使用 Live CD 版本的 Ubuntu 来体验一下 Ubuntu 带来的快感。Live CD 版本是一个在光盘中运行的 Ubuntu 版本，无须安装就可以使用。

Ubuntu 给用户提供了多种安装方式：光盘启动安装、硬盘启动安装、虚拟机安装、网络安装等。

1. 光盘启动安装

Ubuntu 的光盘启动安装是比较普遍的一种安装方式，也是用户选择比较多的一种。在上



一节中，我们讲述了怎么刻录一张 Ubuntu 的启动光盘。光盘启动安装过程简单，操作简便。

用户在使用光盘启动安装的时候，需要先设置计算机的 BIOS。对于台式机，一般来说，在启动计算机的时候，按 F2 或是 Del 键就可以进入 BIOS 的后台设置界面；对于手提电脑，普遍也是按 F2。具体不同的主板，用户可以在网络寻找相应的按键，在这里就不详述了。在 BIOS 中，找到高级设置，一般来说都会有 First Boot 等字样，设置 First Boot 为从 CD 启动。退出 BIOS，保存 BIOS 设置。插入 Ubuntu 启动光盘，计算机就会从启动光盘中引导计算机安装 Ubuntu。

2. 硬盘启动安装

Ubuntu 硬盘安装方式，是针对没有光驱的用户或是不使用光盘的用户提供的一种 Ubuntu 安装方式。硬盘安装也有两种：一种是把 Ubuntu 作为一个应用程序安装到 Windows 系统中，也就是说，不用重新格式化磁盘，就可以在 Windows 中安装 Ubuntu。另外一种是使用 GRUB4DOS 等软件，设置启动 Ubuntu 的引导安装程序。

前者使用了 Wubi 应用程序。Wubi 程序是在 Ubuntu 8.04 中开始得到应用，该功能的最大优势就是允许用户把 Ubuntu 系统安装到 Windows 操作系统上，而不强制要求独立的硬盘分区。Ubuntu 不影响任何 Windows 操作系统设置，用户可以像卸载任何 Windows 应用程序一样卸载 Ubuntu 系统。

后者则是使用 GRUB4DOS 等引导软件来完成安装。GRUB4DOS 可以从 <http://download.gna.org/grub4dos/> 获得。GRUB4DOS 使用简单，而且是一个开源的软件，符合 GNU。下载 GRUB4DOS 后，解压缩后就可以了。核心文件有下面几个：grldr 和 grub.exe 及 menu.lst（此为样本，需修改）。如果你需要中文支持，就用 chinese 子目录中的文件。

硬盘安装 Ubuntu 系统，在第 5 章有详细的说明。

3. 虚拟机安装

这种方式，是在 Windows 下先安装一个虚拟机软件，如 VMware。一般用户如果想尝试安装 Ubuntu Linux，最直接的方法就是在一台计算机上安装多个操作系统，但是在一台计算机上安装多个操作系统常常容易出错，而且容易导致硬盘错误，那样的话所有系统可能将全部报销。而在虚拟机中用户可以翻江倒海任意妄为，而几乎不会影响原计算机系统的硬盘数据。

虚拟机安装给用户提供一个尝试的平台，使用 Ubuntu 的同时还可以使用 Windows 系统，多种操作系统同时存在。在虚拟机上操作 Ubuntu 与操作安装的硬盘中的 Ubuntu 是一样的。可是使用虚拟机来运行 Ubuntu，用户就不用担心 Windows 系统下的数据会丢失，或破坏 Windows 系统。具体的安装方法，在第 5 章会阐述。

4. 网络安装

只要可以连接到网络，就可以使用这种方式来安装 Ubuntu 系统。Ubuntu 是集成 Debian 系统的。因此，Ubuntu 也可以从网络上安装到硬盘中。这种方式方便快捷，只要对网络进行一番配置就可以了，还需要配置数据源。

3.3.3 规划磁盘分区

首先我们需要知道，硬盘分区的存在，是由硬盘的物理特性决定的，并不会因为不同的

操作系统而有所改变。Windows 下的文件系统和 Linux 的文件是不一样的。Linux 没有盘符这一概念。Linux 的磁盘分区是通过挂载来完成的，目录形成树状结构。

我们来看看 Linux 和 Windows 对于分区不同的表示方法：

可能您已经很熟悉 Windows 了，它使用盘符来表示分区，比如 C: D: E:，每一个分区使用一个盘符来标识，而且顺序可以颠倒，D:并不一定就是您系统中的第二个分区。

而在 Linux 中，分区是这样表示的：

```
/dev/hda  
/dev/hda1  
/dev/hda2  
/dev/hda5  
/dev/sdb1
```

以 /dev/hda5 为例，因为在 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，我们来看剩下的部分 hda5。前两位的字母 hd 表示这是一块 IDE 硬盘，如果是 sd，则代表 SATA 硬盘，或者闪存等外设。第三位的字母 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二、三、四个设备。例如 hdc 表示第二个 IDE 接口上的主硬盘（每个 IDE 接口上允许一个主设备和一个从设备）。第四位的数字 5，并不表示这是该硬盘中的第 5 个分区，而是第一个逻辑分区。因为在 Linux 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硬盘中的位置决定。系统又要为所有可能的主分区预留标识，所以 1~4 一定不会是逻辑分区，5 则是第一个逻辑分区，以此类推。Linux 下面每一个硬盘总共最多有 16 个分区。

了解了分区之后，我们就要来看看这些分区是怎么在 Ubuntu 下使用的。我们需要先来说一个重要的概念——“挂载”。

在 Windows 操作系统中，挂载通常是指给磁盘分区（包括被虚拟出来的磁盘分区）分配一个盘符。这个操作可以通过“计算机管理”中的“磁盘管理”来进行。第三方软件，如磁盘分区管理软件、虚拟磁盘软件等，通常也附带挂载功能。在 Linux 操作系统中，挂载是一个非常重要的功能，使用非常频繁。它指将一个设备（通常是存储设备）挂接到一个已存在的目录上。这个目录可以不为空，但挂载后这个目录下以前的内容将不可用。

需要理解的是，Linux 操作系统将所有的设备都看作文件，它将整个计算机的资源都整合成一个大的文件目录。

我们要访问存储设备中的文件，必须将文件所在的分区挂载到一个已存在的目录上，然后通过访问这个目录来访问存储设备。

我们需要分开分区和目录的概念。Ubuntu 下的目录，相当于一个指针或者说是一个写有编号的纸条，分区是一个存储空间或是带有编号的用来存放物品的“房子”。分区和目录是通过挂载的操作连接在一起的。

Ubuntu 下有几个比较重要的目录，如根目录“/”、“/home”、“/user”、“/var”。还有一个专门用于保存启动配置的“/boot”目录，这个“/boot”是比较重要的，只要这个目录的文件没有被损坏，系统还是可以启动的。一般为了简单起见，我们把“/boot”和“/”放在一起。



grub 启动时需要的文件都在 /boot 目录。这样就算工作分区出了问题，只要这个分区没有问题，同样可以启动。之后的分区全部以扩展分区形式存在，扩展分区下有很多逻辑分区。

“/usr”里存放有大部分的系统软件，包括图形界面程序，其需求的磁盘空间也是很大的。

“/home”的挂载分区，主要是用来保存用户的工作文件，多人使用，为多人的工作文件保存区。因此，这个分区是越大越好，没有一个上限。“/home”保存许多用户的信息，包括用户的启动配置，软件的使用配置等。

其他的诸如 tmp, var 等来说，单用户工作机是不用考虑的。没有分配磁盘空间的 Ubuntu 目录，一般都会挂载到 “/” 中。

Ubuntu 有一个交换分区，这个交换分区的大小一般为内存的一倍到两倍。分区格式改成 Linux-swap。

对于初学者来说，一般为 Ubuntu 分配两个分区就可以了，在扩展分区上分配给 “/” 4G 以上的磁盘空间，在逻辑分区上分配给交换分区一个大小为内存的一倍到两倍的磁盘空间。

3.4 注意事项



尽管 Ubuntu 的安装很简单，但是 Ubuntu 社区也不能完全考虑到每个用户的要求。因此，还是在读者安装 Ubuntu 的同时难免会遇到各种问题，这里列举了在安装 Ubuntu 时需要注意的常见问题。

3.4.1 下载映像文件

现在映像文件操作很简单，我们要选择比较近的资源库下载点，默认的欧洲下载点。一般来说，这个软件源下载的速度比较慢，我们需要选择比较近的软件源，一般选择日本或我国台湾的资源点，这样下载速度比较快。可以到中文的 Ubuntu 官方网站下载，网址为 <http://www.ubuntu.org.cn/getubuntu/download>。

3.4.2 硬盘分区

硬盘分区是一个比较复杂的难题。若你还在使用 Windows，你可以使用 Windows 下的分区魔术师来进行分区。尽量为你的 Ubuntu 腾出 10GB 左右的硬盘空间。格式化硬盘时，设置为 ext3，或是其他的 Linux 支持的文件系统格式。

若你之前的硬盘已有 3 个主分区，建议你最好不要创建一个主分区。那样会破坏你计算机中的其他操作系统。第一个逻辑分区，还是可以挂载你自己的文件目录，这样就可以保护你的文件，就算 Ubuntu 系统崩溃了，你的数据还有，只要在下次安装 Ubuntu 的时候把这一分区重新挂载就可以了。为你的 “/home” 目录分配一个分区是很有必要的。

格式化硬盘时，一定要确定你没有格式错分区，对很重要的数据，建议还是做一个备份为好。

3.4.3 网络配置

在不清楚你的网络连接的情况下，建议先完全安装 Ubuntu 系统之后，再更新软件库。安装之后可以通过设置软件源，以达到快速更新的效果。因而在安装 Ubuntu 的时候，你可以拔掉你的网线，加快安装的速度。

若你使用的是 Live CD，就可以清楚地了解到你的网络情况，并且设置网络配置，通过 Live CD 系统来指引 Ubuntu 的安装。

3.5 小结



这章简述了安装 Ubuntu 的基本准备工作。了解 Ubuntu 对硬件系统的要求，下载映像文件，刻录启动光盘，记录硬盘的相关信息，针对具体情况选择合适的安装方法。在最后的一节里，指出了在安装的过程中需要注意的几点。

Ubuntu 的安装方式有多种，通过光盘启动、通过硬盘启动，还可以通过网络启动。Ubuntu 提供了方便的安装，适合普通用户。

在第 4、5 章，我们将详细简述怎样安装 Ubuntu 9.04。第 4 章讲述的是 Ubuntu 9.04 的光盘安装方式，第 5 章讲述的是 Ubuntu 9.04 的硬盘安装和虚拟机安装方法。

3.6 习题



1. 安装 Ubuntu 有哪几种方法？
2. 给自己的硬盘计划一个合适的分区方案。
3. 按照 3.2.2 节的说明刻录一张 Ubuntu 启动盘。

Linux

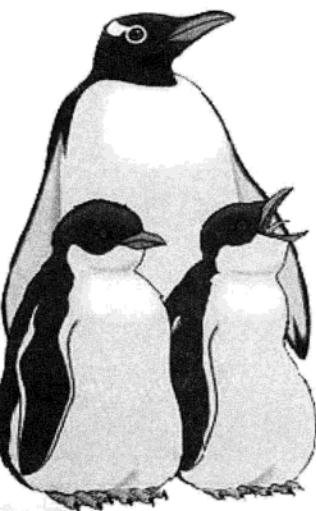
第4章

单系统安装 Ubuntu

在前一章，我们讲述了安装 Ubuntu Linux 需要做的预备工作。读者应该有一张已刻录好的 Ubuntu 启动光盘在手上了，如果还没有刻录 Ubuntu，可以按照第 3 章的方法为自己刻录一张，并且已经确定了在哪里安装 Ubuntu，而且也应该为 Ubuntu 留下了可以用的磁盘空间。接着，我们就要具体讲述怎么把 Ubuntu 安装到你的硬件上了。

如果你还没有用过 Ubuntu 或是 Linux 的操作系统，则你可能对安装程序提出的有些问题感到很陌生，但实际上 Ubuntu 的安装是十分简单的，可以说是“傻瓜式”的安装，许多较困难的决策已经由安装软件完成，如硬盘分区，挂载分区。

Ubuntu Linux 支持多种安装方式：CD 安装，硬盘安装，虚拟机安装，网络安装。本章，我们主要讲述的是怎样在你的计算机上安装单一的 Ubuntu Linux 系统，在后续章节将讲述其他安装方式。





4.1 启动 Live CD

目前，我们已具备在硬件系统上安装 Ubuntu Linux 的条件：拥有一份自己系统的配置清单，一张已刻录好的 Ubuntu Linux 启动光盘，安装系统的硬盘空间等。现在开始安装 Ubuntu Linux。之前说过，安装 Ubuntu Linux，可是说是“傻瓜式”安装，也就是说对普通用户来说，只要你按照安装步骤，不做挑战的行为，你完全可以顺利地完成安装 Ubuntu Linux 的旅程。

下面来体验一下 Ubuntu Live CD 带来的方便，使用光盘引导 Live CD 是很简单的事情，单击“Next”就可以。

第一步，理所当然是启动计算机，接下来就是在 BIOS 中设置光盘启动计算机。设置光盘启动计算机的方法，在这里不做描述。插入 Ubuntu Linux 启动盘，就可以开始安装 Ubuntu 程序。在安装的过程中，Ubuntu 的引导程序会一步一步地教导读者怎么完成安装，而且提供了比较智能的解决方案。

插入已刻录好的 Ubuntu 光盘，你将看到第一幅画面，按下 F2 选择语言：如图 4-1 所示。在这里 Ubuntu 提供了很多国家的语言，你可以使用英语，日语，韩语，还有我们的国语简体中文，也可以选择繁体中文。对我们大陆的读者来说，我们还是选择简体中文了，移动方向键，使颜色条指向简体中文，按下回车键完成语言选择。

看到了图 4-1 所示的这个画面，表示你已踏入了安装系统第一步了。选择中文（简体），在这里选择语言，对我们安装 Ubuntu Linux 之后的系统是很有帮助的，这样系统可以很好地支持中文，而且系统会自动给你配置一个舒适的中文应用环境。

选择中文（简体）语言之后，你会看到系统安装的提示画面，如图 4-2 所示。

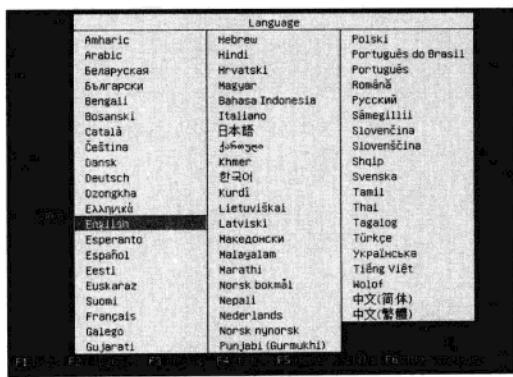


图 4-1 选择语言

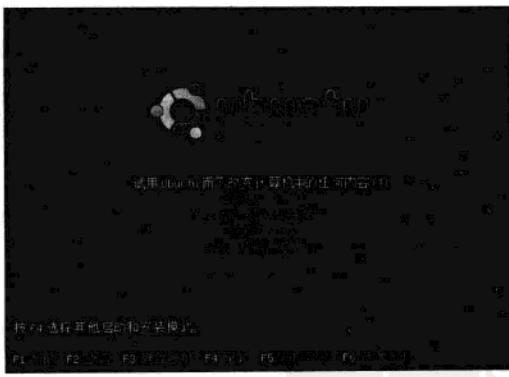


图 4-2 安装模式提示

在这一界面中，可以看到有五个选项“试用 Ubuntu 而不改变计算机中的任何内容 (T)”，“安装 Ubuntu (I)”，“检测光盘是否损坏 (C)”，“测试内存 (M)”，“从第一块硬盘启动 (B)”。很显然，这些选项的意识简单明了。选项一，就是启动光盘上的 Ubuntu 系统；选项二，表示直接安装 Ubuntu 系统到磁盘中；选项三，表示检测光盘是否完整，检测是否有受损，光盘受



损会影响安装，严重的话，会使你无法成功安装 Ubuntu 系统；选项四，内存测试，测试你的内存是否没有问题，检查内存空间是否足够安装 Ubuntu 系统；选项五，大家都知道是怎么回事，在 Windows 系统中是比较常见的选项。

我们要进入 Live CD，因此在这一步，选择第一选项“试用 Ubuntu 而不改变计算机中的任何内容”，以引导进入 Ubuntu Linux Live CD 系统。按回车键，Live CD 系统开始引导启动，效果如图 4-3 所示。



图 4-3 引导 Live CD

此引导过程会花费你几分钟的时间，引导完成后，系统会以默认用户登录到 Live CD 系统，你现在就可以使用 Ubuntu Linux 系统。Live CD 界面如图 4-4 所示。

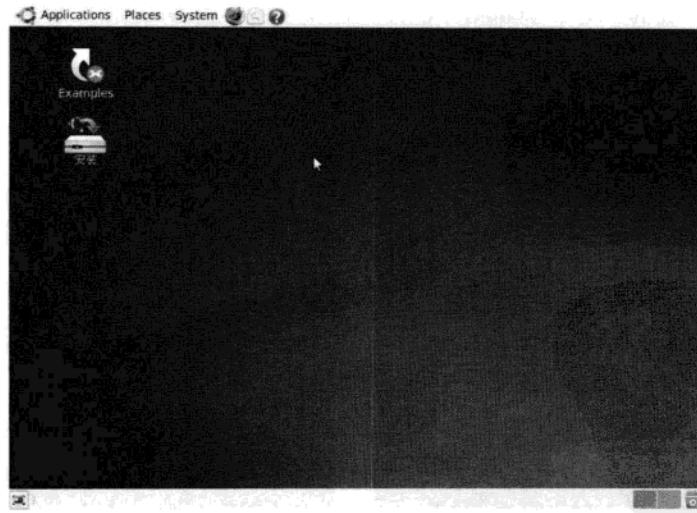


图 4-4 Live CD 系统界面

Live CD 系统是一个在指定的磁盘空间上运行的 Ubuntu Linux 系统。这里以光盘为 Live CD 的运行空间。在 Live CD 系统下你可以体会到 Ubuntu 的神奇，这系统具有安装版的基本操作功能，几乎和安装 Ubuntu 一样。成功地进入了 Live CD，接下来就要为我们的硬盘进行分区。



4.2 在 Live CD 中磁盘分区

也许你的计算机硬盘是刚买回来的，还没有进行格式化；也许你的硬盘之前安装过 Windows 或其他系统，硬盘格式为 Linux 不支持的文件系统格式。现在你要在你的硬盘上安装一个全新的操作系统，你明智地选择了 Ubuntu Linux 为你攻入 Linux 世界的第一操作系统。那你就要先对你的硬盘进行分区。在准备安装的章节，我们简述了分区磁盘的基础知识。Live CD 系统中文显示不是很好，有些菜单项没有对应上，这个不用担心，当你安装完整的 Ubuntu Linux 之后，你就可以更新语言支持库，那会给你一个完美的中文界面。

4.2.1 运行 Partition Editor

在这里我们将详述在 Live CD 中使用系统附带的磁盘分区工具 Partition Editor。通过选择“System” | “Administration” | “Partition Editor”选项（如图 4-5 所示），打开分区编辑软件。

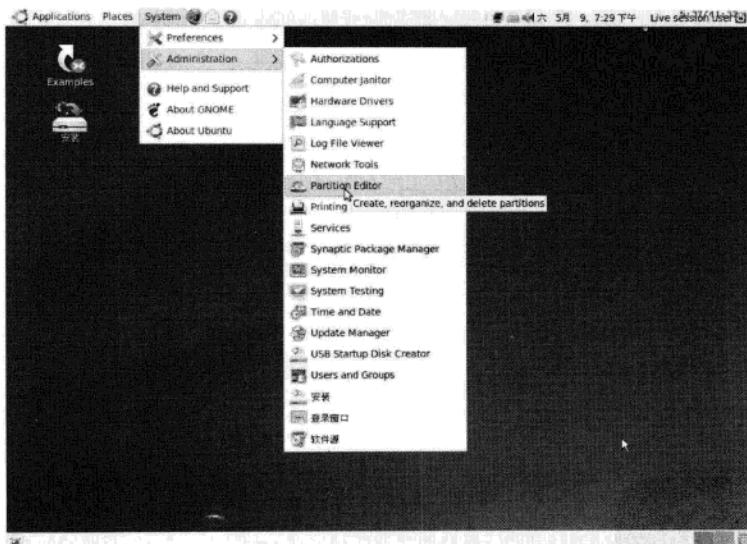


图 4-5 打开 Partition Editor

使用 Partition Editor 可以很方便地对你的硬盘进行分区。接下来，我们就说说怎么使用 Partition Editor 对硬盘进行分区。打开 Partition Editor 的界面，如图 4-6 所示。在 Partition Editor 中，我们要删除所有的分区，如果你还有一些必要的文件保存在 Windows 文件系统格式，也可以选择部分分区来完成 Ubuntu 的安装。在这里，我们使用了一个 8GB 空间的硬盘安装 Ubuntu 系统，通过对这 8GB 的磁盘空间来讲解安装 Ubuntu 的磁盘分区工作。

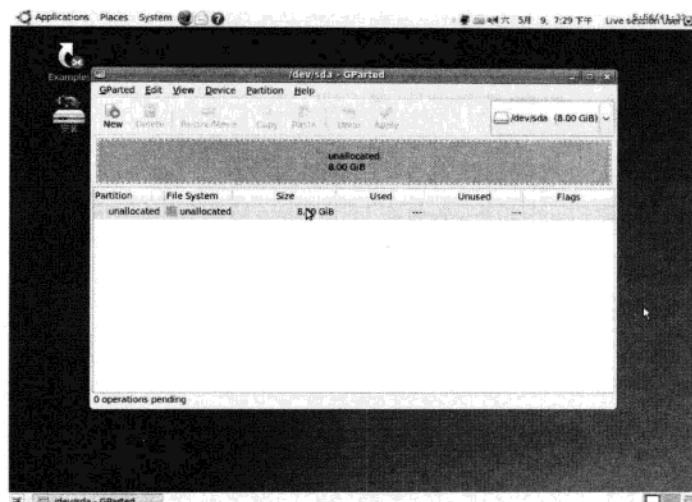


图 4-6 Partition Editor 界面

打开 Partition Editor 后可以看到我们拥有一个没有格式化的硬盘，在 Ubuntu 下硬盘对应的文件为 /dev/sda。显示为 sda，是因为硬盘为 SATA 类型，之前的章节我们讲述过这一点。接下来，我们就要对这一磁盘进行操作。

单击选择 /dev/sda 分区，选中分区，再选择 Partition | New 或是用鼠标右键单击分区，在弹出的快捷菜单中选择“New”选项，如图 4-7 所示。

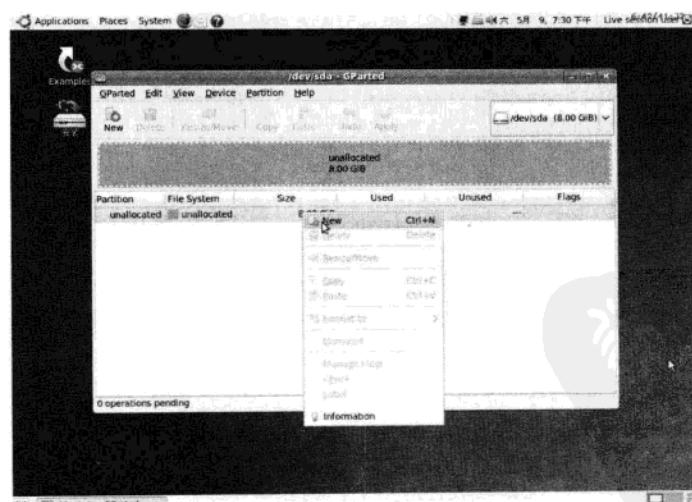


图 4-7 创建新分区

由于这个磁盘是没有经过格式化的，所以会弹出一个警告窗口，如图 4-8 所示。

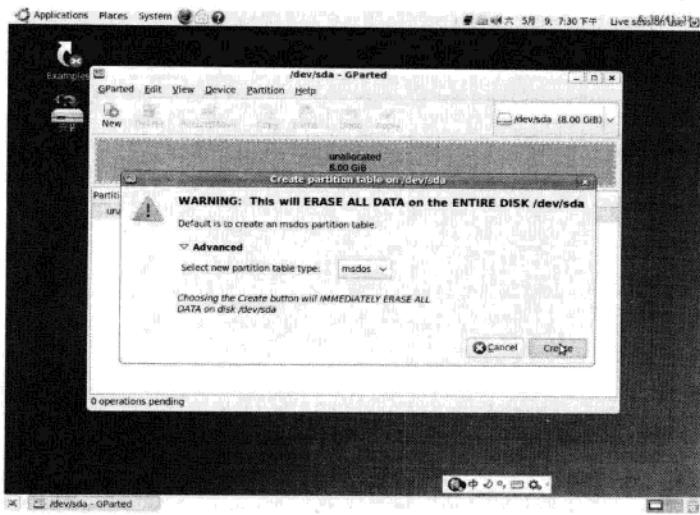


图 4-8 调整分区大小

默认的使用 msdos 的分区表结构，保持这一设置，单击 Create 按钮，创建新的分区表结构。创建新分区需要再次用鼠标右键单击空分区，从弹出的快捷菜单中选择“New”选项，如图 4-7 所示，接着进入划分磁盘分区的操作。

4.2.2 划分磁盘分区

先说说分区策略，这一步骤，请参考第 3 章的说明。此处，我们选择为 /boot 设置一个分区，分配一个交换分区和一个根目录 “/” 分区。为 boot 分配 1GB 的空间，为交换分区分配 1GB 的空间，剩下的磁盘空间就分配给 “/” 目录。boot 分区保存着系统启动时需要的文件，Ubuntu 9.04 启动的时候，都会从这一目录下读取文件。单独分开来是有好处的，就算系统崩溃，也是损坏了挂载 boot 的分区，对其他的分区没有影响，重新安装的时候，把其他分区重新挂载到相应的目录下就可以使用原来的数据，这样分区比较安全。

先创建一个 1GB 的主分区 /dev/sda1，这一分区存放 Ubuntu 系统启动文件和配置文件。这属于创建新分区的操作，操作如上图 4-7 所示。这一次操作之后会弹出如图 4-9 所示的创建新分区的对话框。

设置大小，可以通过拖动白色区域的箭头自由调整到 1GB 的大小，也可以在 New Size (MB) 的选项中设置新建分区的大小。“Create as”有三个选项：一个为主分区 (Primary Partition)，一个为逻辑分区 (Logical Partition)，还有一个就是扩展分区 (External Partition)，此处，我们需要选择 Primary Partition。File System 这一选项列出了许多 Linux 支持的文件系统，如 ext2, ext3, ext4, fat32。尽管说 Ubuntu 9.04 支持 ext4，可是官方发布的时候，还是没有把文件系统默认为 ext4。为了系统的稳定性，我们还是选择了 ext3 作为 Ubuntu 的文件系统。单击“Add”按钮，完成创建新分区。

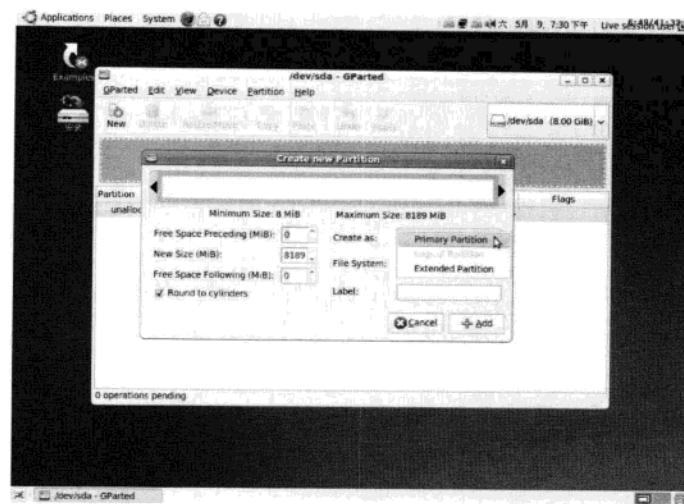


图 4-9 为 /boot 创建一个分区

创建了一个主分区，我们还需要一个扩展分区来为我们工作。把剩下的所有磁盘空间分配给扩展分区。有读者就会问，不是说要分 3 个分区吗？怎么现在只有 2 个分区？不要着急，之前我们说过，在 Linux 下有 3 个比较接近的概念：主分区，扩展分区，逻辑分区。逻辑分区是在扩展分区上划分的。创建步骤同上，如图 4-10 所示。

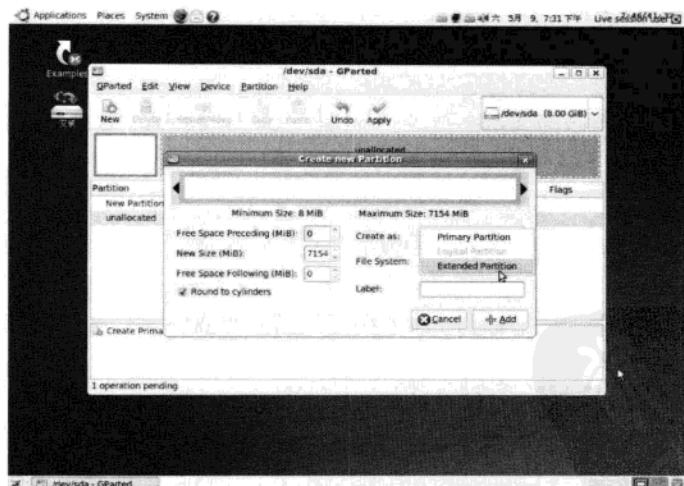


图 4-10 创建扩展分区

剩下的磁盘空间都分配给扩展分区，因此在这一界面我们只要在“Create as”下选择“External Partition”选项就可以了。单击“Add”按钮完成创建扩展分区任务。这样我们就把一个磁盘划分为一个主分区和一个扩展分区。

再在扩展分区上划分交换分区和根目录挂载的分区。

创建新分区，用鼠标右键单击扩展分区选择新建菜单。出现如图 4-11 所示界面。

理论上，交换分区的大小通常是物理内存的 1~2 倍。在“Created as”下选择“Logical Partition”选项；而在“File System”下选择“Linux-swap”；Label 部分留空；选中“Round to cylinders”复选框；单击“Add”按钮完成设置。完成交换分区的划分。

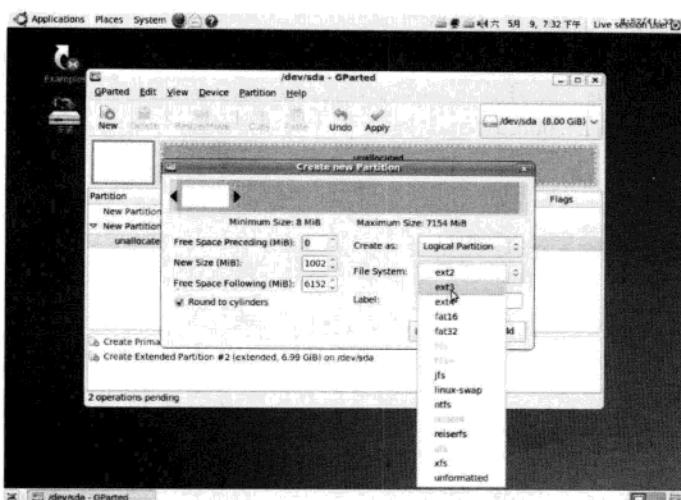


图 4-11 创建交换分区

剩下的磁盘空间还有大约 6GB，我们把这一部分分区分给根目录分区。创建操作同上，Create as 选项会自动选择“Logical Partition”选项，在“Filesystem”目录下选择“ext3”选项，单击“Add”按钮完成磁盘的划分。完成分区的结果如图 4-12 所示。

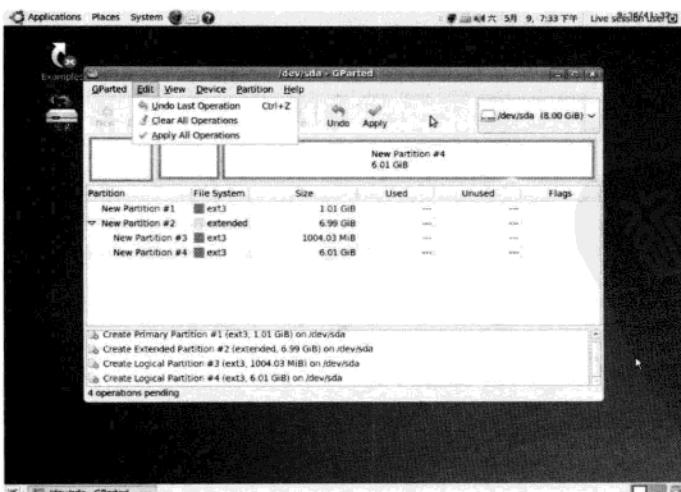


图 4-12 磁盘划分结果

到这里，分区的所有操作就完成了，可是，计算机还是没有对这些操作进行处理。确定



分区无误，我们就可以选择“Edit”|“Apply All Changes”选项，或是单击工具栏上的对钩符号，应用我们的分区方案。软件会提示，你是否真的执行操作，只要按“确定”就可以了。接下来的工作就交给软件去完成。还有一个应该注意的，为了硬盘上数据的安装，无论是安装还是格式化，都要对你的数据进行备份，在 Partition Editor 执行的时候，最好不要单击取消命令，除非你觉得无所谓。Partition Editor 执行分区操作需要花费一点时间，具体时间要看你所操作分区的大小，分区是否还有其他文件等因素。Partition Editor 执行完操作后，会弹出一个对话框，提示是否需要保存操作记录。你可以保存，也可以选择取消保存，直接关闭对话框。现在已经为我们的 Ubuntu Linux 提供了一个很好的生存空间，退出 Partition Editor。

4.3 安装系统步骤



使用 Live CD 安装 Ubuntu 系统是一件很简单的事情。在 Live CD 系统的工作桌面上有一个安装的图标。安装过程很简单，用户只要按照提示就可以完成安装，对于安装的每个步骤，我们将逐一详细地进行简述，读者只需拖着鼠标在引导安装程序上操作。可以单击安装图标，开始把 Ubuntu Linux 安装到你的硬盘上。

4.3.1 选择语言

双击安装图标，进行安装，如图 4-13 所示。从这一步开始，将由安装引导程序来为我们服务。



图 4-13 选择语言

从这里，我们可以看到，系统已经为我们选择了中文（简体）。这是由于在启动 Live CD 系统的时候，我们选择了中文（简体），自然在这里系统自动识别设置。

4.3.2 选择时区

在一般情况下，安装程序能够正确地识别系统的时区，如果没有准确识别，请在“所选城市”下拉菜单选择所在的城市为 Shanghai，对于大陆来说，系统默认都是 Shanghai，你可以选择其他城市。事实上保持默认就可以了。大陆使用的是北京时间 GMT+8。单击“Forward”选项进入下一步。

4.3.3 键盘布局

国内绝大多数人都使用美式英语键盘，系统会默认你的选择为 China-China。这里可以设置键盘的布局，安装系统之后，也同样可以修改键盘布局的设置。这里使用默认的配置就可以了。

4.3.4 预备硬盘空间

通过之前 Live CD 的分区操作，在这里我们就方便多了，引导程序扫描硬盘的分区情况，并在界面上显示出来，如图 4-14 所示。

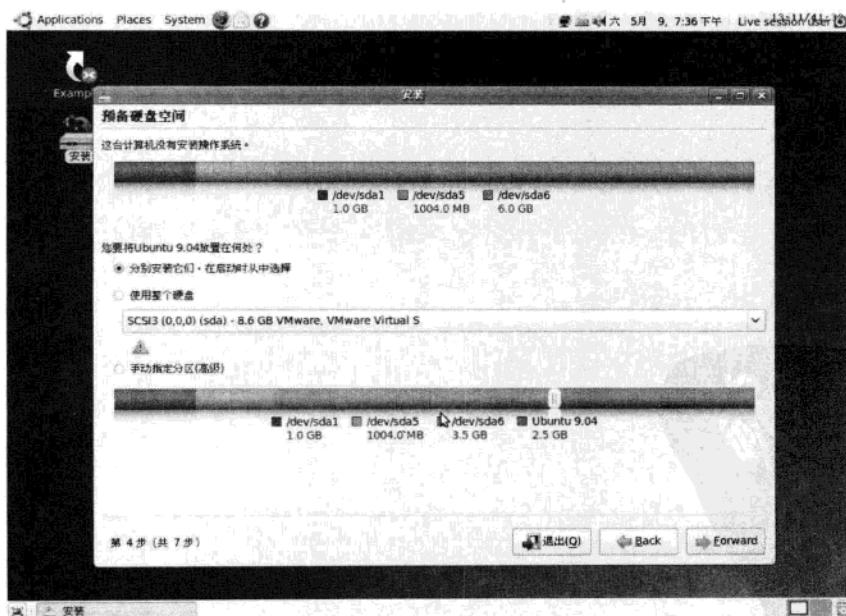


图 4-14 预备硬盘空间

先来解释一下这一步出现的以下选项：

(1) 分别安装它们，在启动时从中选择

系统安装的时候默认此选项，即使用分区向导。使用分区向导的方法很简单，只有两步，



首先分区工具从你的某一个磁盘上划分出一定大小的空闲分区，然后把所有的工作都交给软件来完成。进入下一步，你可以调整用来安装系统的磁盘空间大小，这一操作不会给你在其他磁盘的数据造成影响，不过还是建议你要备份重要数据。接下来，按照提示操作就可以，一切操作都是由分区软件来完成的。

(2) 使用整个分区

系统自动识别出你目前硬盘上文件系统格式为 ext3 的磁盘空间，你可以选择整个分区来安装 Ubuntu 系统。

(3) 手动指定分区（高级）

手动分区，我们可以自行选择分区挂载的目录。

我们为 Ubuntu 创建 3 个分区，一个是启动分区 sda1，一个是交换分区 sda5，一个是挂载根目录的分区 sda6。现在要把根目录分区和根目录联系在一起。选择“手动指定分区”选项，单击“继续”按钮进入下一步“准备分区”。

4.3.5 准备分区

在图 4-15 中，可以看到我们之前划分的磁盘分区：sda1，sda5，sda6。其中 sda1 为主分区，我们打算用来挂载“boot”目录的，sda5 为交换分区使用，sda6 用来挂载目录“/”。在这一步中需要完成创建磁盘分区与系统目录之间的关系。单击或双击分区 sda1，也可以通过单击左下角的“编辑分区”按钮编辑分区，执行操作后，弹出编辑分区对话框，如图 4-16 所示。



图 4-15 准备分区



图 4-16 编辑分区

在对话框中，把“用于”设置为“Ext3 日志文件系统”，把“挂载点”设置为“/boot”。若你自己在操作 Live CD 系统的时候，给你的 Ubuntu 系统划分了几个分区，你可以相应地在这里设置分区与目录的对应关系，挂载点有“/home”，“/user”等。你有自己的分区方案，可以设置它们如设置“/boot”挂载点一样。选中“格式化”复选框，单击“Ok”，退出编辑分区。由于之前我们准备了一个交换分区，这里系统会自动检测到这个分区，设置交换分区的挂载点的时候，只需在“用于”选项上选择“交换空间”选项。

分区配置完毕，我们可以单击“继续”按钮，进入安装的下一步“创建用户”。

4.3.6 创建用户

这一步的标题为“你的身份是什么？”，如图 4-17 所示。在这一步骤，我们需要做的是设置登录账号和密码。对于 Ubuntu 来说，用户是必然要建立的，作为网络操作系统，多人共享的操作系统，Ubuntu 需要使用用户的名称和密码来保证用户数据之间的通信和安全。此处有四个选项，第一个选项要求输入你的名称，第二个选项要求你输入你的登录名称，第三个选项则要求输入你的密码和确认你的输入密码，第四个选项要求你输入一个标识，即计算机的名称。Ubuntu 的安装就是方便大众，操作简单，你输入第一个参数为“man”的同时，第二和第四选项就相应地设置为“man”、“man-desktop”。你需要做的就是设置你的密码和输入确认密码。在 Ubuntu 8.10 的发布版本之后，在界面的下边添加一个自动登录的选项。你若喜欢在不输入的前提下直接登录到系统中去，那你可以在这个选项上打个钩。“自动启动”使用方便，不过，安全性自是没有需要输入密码方式高。

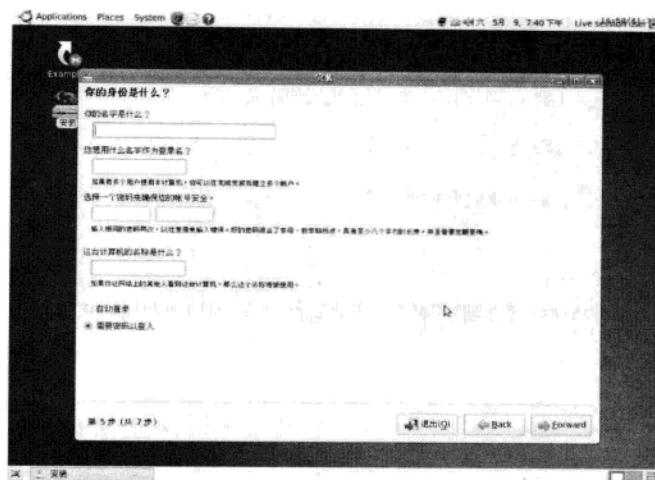


图 4-17 设置登录账号和密码

4.3.7 迁移向导

细心的读者会发现，这种方式安装会少了一步，也就是没有所谓的第 6 步。这一步，其实把你的 Windows 系统下的收藏夹和个人文件都转移到 Ubuntu 下。Ubuntu 系统都默认不迁移文件。保持系统默认，单击“继续”按钮进入下一步。

4.3.8 安装系统配置清单

到此，我们走到了安装系统的最后一步，如图 4-18 所示。

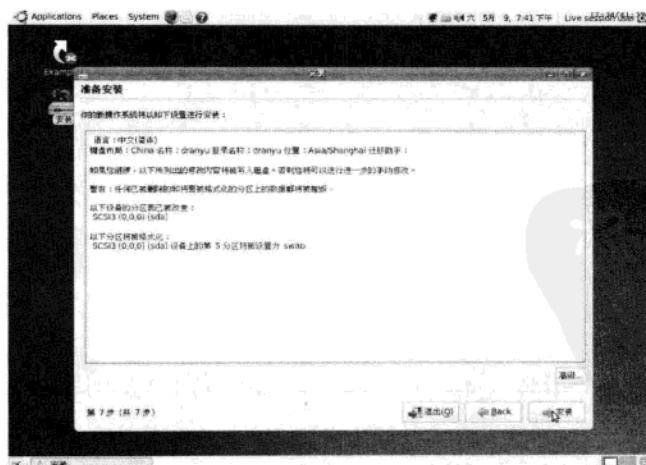


图 4-18 准备安装

在这里，安装向导显示了系统安装的一份清单。详细地阅读，查看是否完全按照自己的意愿去安装。你觉得不满意，完全可以单击“后退”按钮，退到前面的几个步骤，完成你想

要的设置。对照清单，确定无误之后，你就可以单击“安装”按钮进行安装。

接下来的操作都是由系统自动去完成。你可以休息一会儿再回来，系统会提示你，安装已经完成，会弹出如图 4-19 所示的界面。你只要单击“现在重启”按钮就可以完成安装过程了。

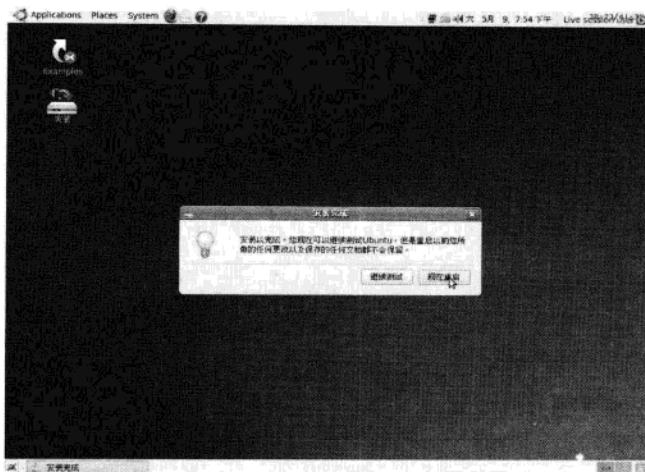


图 4-19 安装完毕

到这里我们的整个安装过程就结束了。可以开始你的 Ubuntu 的体验旅程了。



4.4 小结

本章简述了 Live CD 的启动和在 Live CD 上使用 Partition Editor 为磁盘分区，最后详细说明了 Ubuntu Linux 的安装步骤。每个安装步骤，我们给出详细的解说。基本上在鼠标和键盘的选择上都可以很简单地配置，主要是硬盘分区的配置，我们就需要注意，推荐的分区，有时候不符合我们的要求。在创建用户界面上，要牢记自己的用户名和密码。系统安装之前，引导程序列出安装系统配置清单，查看清楚配置之后就可以安装系统。

Ubuntu 9.04 除了提供光盘启动安装系统之外还提供了从硬盘上启动安装系统。在第 5 章，我们就讲述两种从硬盘中安装 Ubuntu 系统的方法，还讲述一种在虚拟机上安装 Ubuntu Linux 的方法。



4.5 习题

1. 使用 Ubuntu Live CD 有什么用途？
2. 安装 Ubuntu 有几个步骤？包括哪些？
3. 操作：在 Live CD 中，为磁盘分区；安装 Ubuntu 到硬盘上。

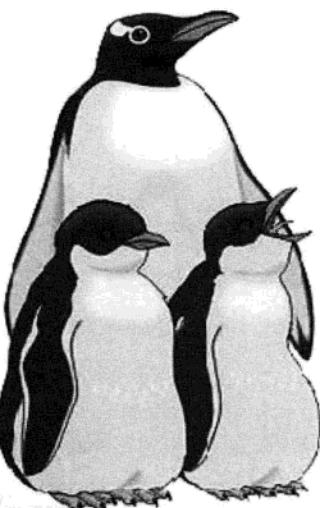
Linux

第 5 章

Windows 下安装 Ubuntu

这一章将讲述在 Windows 系统下安装 Ubuntu Linux。目前对我国大陆来说，桌面系统多为 Windows，要人们在短时间内从 Windows 下转到 Linux 是比较困难的。也就是说，Windows 和 Linux 会在相当一段时间并存，或者说长期共存。可是又有不少人想过过瘾，安装双系统是一个不错的选择，也可以通过虚拟机来体验 Ubuntu 带来的快感。

这章具体讲述三种在 Windows 下运行安装 Ubuntu 的方式。Wubi 安装 Ubuntu，使你的安装 Ubuntu 系统的步骤如同安装一个 Windows 软件一样简单，同样简单卸装。GRUB4DOS 安装 Ubuntu，使你在没有光驱的情况下也可以在你硬盘上安装多操作系统。使用虚拟机安装 Ubuntu，使你可以一边在 Windows 下看电视、听音乐，一边使用 Ubuntu 系统。





5.1 Wubi 无痛安装

Wubi 安装 Ubuntu 是一种专门针对 Windows 用户的 Ubuntu 安装方式。Wubi 读起来可以说和我们的“五笔”一样。Wubi 是“Windows based Ubuntu Installer”的缩写。

5.1.1 Wubi 安装的特点

Wubi 原本是一个第三方工具软件，于 2008 年被收入 Ubuntu 官方 iso。它为 Ubuntu 提供了一种独特的安装方式，在 Windows 下像安装一个应用程序一样来安装 Ubuntu 系统。这种方式的好处是：无须专门为 Ubuntu 分区；采用 GRUB4DOS 而非 GNU GRUB 作为引导器，不写硬盘 mbr，对现有系统无影响，出了问题在 Windows 下就可修复。

Wubi 与虚拟机有本质区别。Wubi 安装是在现有的分区中创建磁盘映像文件（虚拟硬盘），将 Ubuntu 系统文件写入其中，启动时将其挂载为根文件系统。一般，根文件系统是由一个或几个分区构成，而这里只是硬盘中的一些文件。这就意味着通过复制这些文件，即可轻松实现系统克隆；将映像文件存入移动硬盘，稍加配置，就变成了可移动的 Ubuntu 系统。

Wubi 安装比较有针对性，主要面向的用户为 Windows 用户。操作简单，在 Windows 下安装 Ubuntu 就像安装其他 Windows 应用软件一样。用户不用考虑磁盘的分配，也不用害怕安装 Ubuntu 会对 Windows 造成损害。安装简单、安全，这就是 Wubi 安装给 Windows 带来的最大特点。

5.1.2 Wubi 安装配置

Wubi.exe 从 Ubuntu 8.04 之后就加入到了 Ubuntu 的发行版本中。它简单快捷的安装特点大受 Windows 用户欢迎。安装 Ubuntu 8.04 和 Ubuntu 8.10 的时候，安装分区的磁盘格式最好为 Fat32，可是到了 Ubuntu 9.04，你必须在 NTFS 文件格式系统中安装，才能安装成功。这是 Ubuntu 9.04 的一点不足之处。

如果你的系统中没有一个分区的格式为 NTFS 格式的，你就必须重新对磁盘分区进行一些修改。在 Windows 下，我们建议使用 PartitionMagic8.0f 软件对磁盘分区进行调整。使用 PartitionMagic 软件，操作简单，而且不会破坏已有的数据，安全有保障。

需要为我们的 Ubuntu 9.04 准备一定的磁盘空间，使用 PartitionMagic 从一个已有分区上划分出一定的磁盘空间，最好有 8GB，然后新建一个分区，并把分区格式化为 NTFS 文件格式。这样我们就准备好使用 Wubi 安装的 Ubuntu 9.04 的条件了。

现在来说说使用 PartitionMagic 的调整分区和创建新分区的简单操作，在 Windows 下安装 PartitionMagic 就不细说了，读者也可以从网上下载绿色版的软件。

启动 PartitionMagic，你可以看到如图 5-1 所示的界面。使用其他版本的 Partition Magic 也同样有相关的菜单功能。

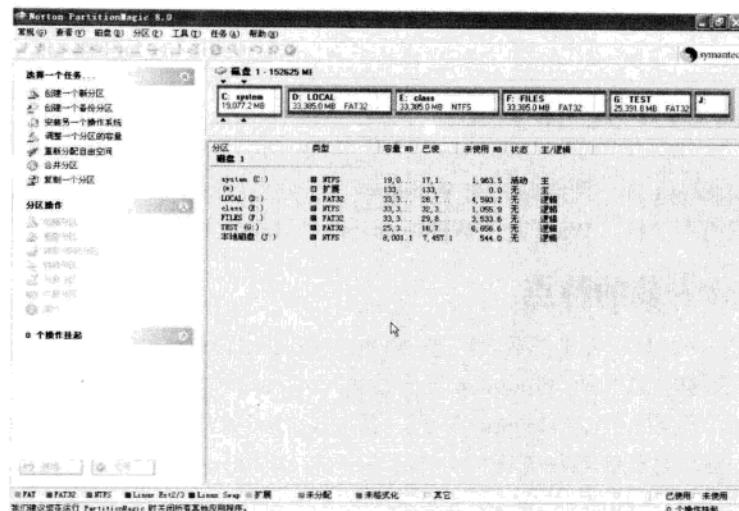


图 5-1 PartitionMagic8 启动界面

在磁盘分区情况条上,可以看到各个磁盘分区的使用情况和相关的系统信息。在 Windows 下磁盘可分为为主分区和逻辑分区,和 Linux 下的概念有所不同。使用鼠标选择你剩余磁盘空间比较大的磁盘分区,从中划分一定的磁盘空间,最少 4GB。在这里,笔者选择了还剩下 5GB 磁盘空间的 G 盘,用鼠标右键单击 G 盘,在弹出的快捷菜单中选择“调整容量/移动”选项,进入调整磁盘空间的对话框,如图 5-2 所示。这里对话框中有一些术语是我们需要清楚的,“自由空间”、“新建容量”和“簇的大小”这些选项。选项“簇的大小”我们没有必要去修改它,“自由空间”表示我们修改分区之后,剩下那一部分为分区的空余空间;“新建容量”表示我们目前使用的原来的分区的大小。

使用鼠标拖动磁盘分区情况栏上左右的两个箭头来修改现有磁盘的分区情况,我们计划在 G 盘的后边创建一个新分区,拖动右边的箭头,待自由空间之后的选项显示为 4000MB 左右的时候停止拖动。单击“确定”按钮完成第一步。

第二步,创建分区。此时你可以看到有一个未分配的分区,分区名为(*)。用鼠标右键单击此分区,在弹出的快捷菜单中,选择“创建分区”选项,弹出“创建分区”对话框,如图 5-3 所示。

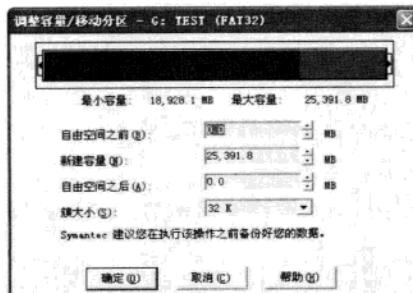


图 5-2 调整容量/移动分区

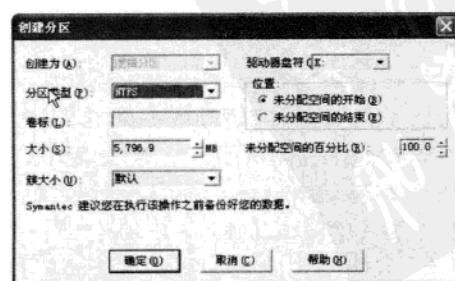


图 5-3 创建分区

Windows 下安装 Ubuntu

按照默认的选项就可以，需要注意的是分区类型一定要指定为 NTFS。这样我们才能够顺利在 Windows 下安装 Ubuntu。单击“确认”按钮之后，回到主界面，如图 5-4 所示。

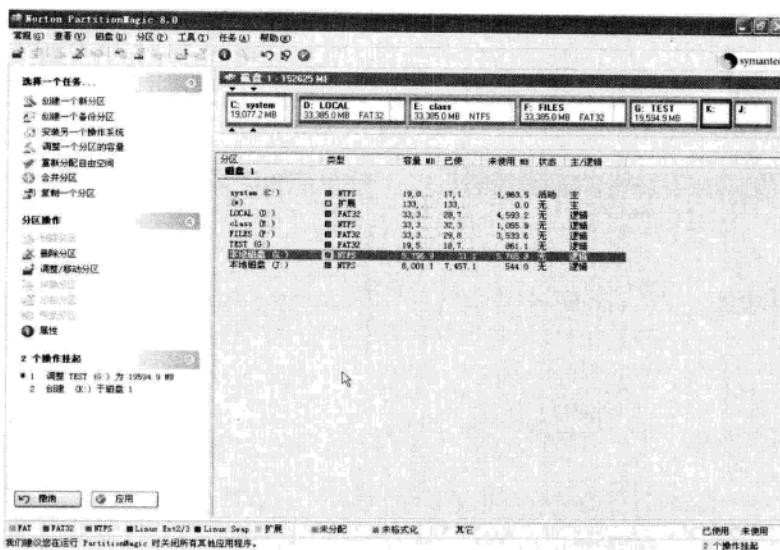


图 5-4 完成分区调整

可以在左下角看到我们需要对磁盘进行的两个操作，单击“应用”按钮就完成了 Wubi 安装的准备工作。

5.1.3 Wubi 安装过程

在上一小节中，我们使用 PartitionMagic 软件为我们安装 Ubuntu 9.04 创建了一个新的分区，把所下载的 Ubuntu 映像文件复制到这一新分区下，我们从 Ubuntu 9.04 的映像文件解压出 Wubi.exe，把它放到 Ubuntu 9.04 的映像文件的目录中。双击打开所下载的 Ubuntu 9.04 映像文件，可以看到如图 5-5 所示的画面。

运行 Wubi.exe，进入 Wubi 的设置界面如图 5-6 所示。

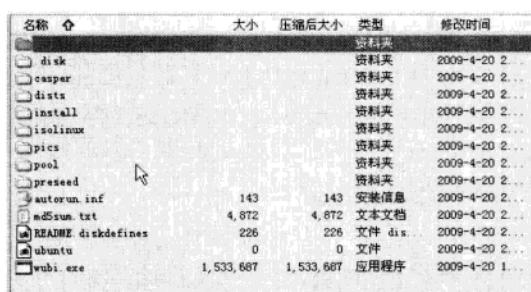


图 5-5 Ubuntu 9.04 映像文件



图 5-6 Wubi 设置界面



在此登录界面上，设置安装的分区符，选择刚格式化的新分区“K:”，在 Installation site 选项下，按下下拉菜单选择允许的最大空间为 4GB，再选择语言，Ubuntu 9.04 下的 Wubi 默认的选项为繁体，我们需要选择“Chinese(Simplified)”选项。在 Username 中输入你登录时的名字，在 Password 下输入你的密码和确认密码。这样就完成了 Wubi 的配置，结果如图 5-7 所示。

单击“Install”按钮就可以进入自动配置安装 Ubuntu 的环境，剩下的事情都是由计算机来完成。可以从数据源站点下载相关的安装信息。如图 5-8 所示。



图 5-7 完成配置

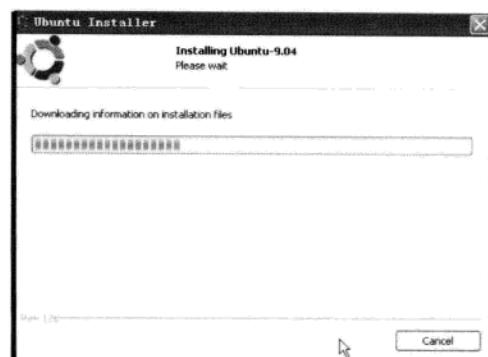


图 5-8 下载安装信息

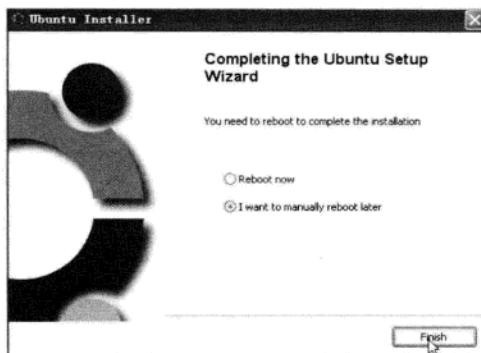


图 5-9 完成 Ubuntu 安装导航

软件自动生成 Ubuntu 文件夹等操作，最后完成配置，弹出如图 5-9 所示界面完成 Ubuntu 的安装。

我们可以选择“现在重启”选项或者“一会再自己重启计算机”选项。目前我们的 Ubuntu 可以说已经安装到了 Windows 下。当你启动 Windows 的时候，你可以在引导程序的多系统选项上看到我们原来的 Windows XP 系统的提示，还有一个我们需要安装的 Ubuntu 提示系统。选择 Ubuntu 进入 Ubuntu 的安装界面。这个 Ubuntu 安装界面是不需要人为干涉的，系统会自动完成安装。这也突出了 Wubi 安装 Ubuntu 的一大特点。

这里需要说的是为什么我们需要把文件系统格式化为 NTFS。Fat32 文件系统格式是不能存储大于 4GB 的文件的，而 NTFS 可以。Ubuntu 9.04 中，Wubi 软件程序只为 Ubuntu 系统创建了一个 root 和 swap 的虚拟空间，swap 交换空间就是我们一般说明 256MB 这对 Fat32 和 NTFS 都没问题。要是我们在设置 Installation Site 的时候选择了大于 4GB 的话，那么必须在 NTFS 格式的分区上安装，要不 Fat32 不能接受 Wubi 所创建的文件。

5.1.4 Wubi 卸装 Ubuntu

有一天你不想使用 Ubuntu 的 Wubi 安装，而改为硬盘安装，或是你需要腾出一些空间来

Windows 下安装 Ubuntu

存放重要的数据的时候，那你可以通过卸装 Wubi 中的 Ubuntu 达到你的目的。Wubi 安装除了安装简单之外，它的卸装也同样简单。进入分区的 Ubuntu 根目录，如图 5-10 所示。

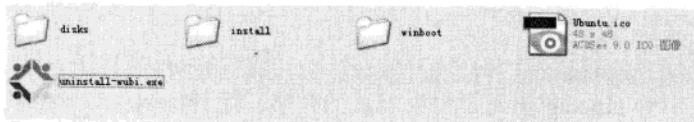


图 5-10 Ubuntu 根目录

双击 `uninstall-wubi.exe`，就会弹出一个提示对话框，询问是否需要卸装 Ubuntu 系统，单击“Uninstall”按钮就 OK 了。

5.2 使用 GRUB4DOS 进行安装



Wubi 安装对 Windows 用户来说是不错的选择，安装方便，操作简单。可是 Wubi 安装也有它不好的一些地方，Wubi 不提供分区，它默认给系统分配了 `root` 分区和 `swap` 分区，对数据的保存有很大的危险性。而且空间比较有限，适合想接触 Linux 的用户。

硬盘安装还可以使用 GRUB4DOS 来引导安装 Ubuntu。

5.2.1 GRU4COX 简介

GRUB 是一个遵从 Multiboot（多重启动）规范的启动管理程序。现在接触到的有 3 种，即 GNU Grub Lagecy , GNU GRUB2 和 Grub for dos (GRUB4DOS)。

GNU GRUB Lagecy 其实就是原来的 GNU GRUB 0.xx，最新版是 2005 年发布的 GNU GRUB 0.97。目前已停止开发，并改名为 GNU GRUB Lagecy。

GNU GRUB2 是第二代 GRUB，它将取代原来的 GNU GRUB（例如 0.9x 版），但目前还处于开发阶段，尚未发布正式版。

GNU GRUB Lagecy 和 GNU GRUB2 都是 GNU 组织的项目。

GRUB for DOS (GRUB4DOS) 是一个以 GNU GRUB 为基础的功能强大的引导器。它可以在 DOS 和 Linux 下运行，也可以通过其他引导器来运行，还可以作为 MBR 运行。GRUB4DOS 内置了功能完善的 BIOS 级磁盘仿真。这里主要讲述 GRUB4DOS。

5.2.2 GRUB4DOS 安装

首先，我们需要下载 GRUB4DOS。可以从以下链接获取到：

<http://download.gna.org/grub4dos/>

<http://download.gna.org/grubutil/>

<http://grub4dos.jot.com/WikiHome>

<https://gna.org/projects/grub4dos/>



下载到最新版本后，解压后可以看到几个关键的文件或文件夹：grldr、grub.exe 及 menu.lst。如果你需要中文支持，就用 chinese 子目录中的文件。安装 GRUB4DOS 也十分简单。

将 grldr 和 menu.lst 文件复制到 c:\，去掉 boot.ini 的只读属性，然后修改 c:\boot.ini 文件，在 boot.ini 的最后面加一行：

```
c:\grldr="Start GRUB4DOS"
```

并将 boot.ini 中的 timeout 值设置为大于 0 的数字，如 timeout=5，然后保存 boot.ini，改回只读属性。

这样，编辑完重新启动计算机，在 XP 的启动菜单就会出现“Start GRUB4DOS”这一项，选择该项即可进入 GRUB4DOS 的环境。

5.2.3 GRUB4DOS 安装 Ubuntu

把 ubuntu-9.04-desktop-i386.iso 复制或移动到 Windows XP 的系统盘从 ISO 文件 casper 目录下的 vmlinuz 和 initrd.gz 文件解压到系统盘。

对于多数没有或很少接触过 Linux 的 Windows 用户来说，刚开始使用 GRUB 时离不开菜单。也就是说，使用 GRUB 前，我们要准备 menu.lst 文件。

menu.lst 文件

```
timeout 3
default 1
title Ubuntu 9.04 install
rootnoverify (hd0,0)
kernel /vmlinuz boot=casper iso-scan/filename=/Ubuntu-9.04-desktop-i386.iso ro quiet
splash locale=zh_CN.UTF-8
initrd /initrd.gz
```

可以重启计算机，这样就进入到 Ubuntu Live CD 系统的界面。剩下的工作就可以如同第 4 章的 Live CD 安装 Ubuntu 一样。

注意：存放 ubuntu-9.04-desktop-i386.iso 的分区格式一定要为 Fat32。这与 Wubi 安装的文件格式不一样。由于 GRUB4DOS 不支持 NTFS。只要能引导进入 Live CD 系统就可以，因此你可以在任意一个 Fat32 分区上存放 ubuntu-9.04-desktop-i386.iso。设置 rootnoverify 行时，需要做相应的修改，如果放在 D:，在 (hd0, 0) 就要改为 (hd0, 5)。

5.3 虚拟机安装 Ubuntu



想要体验 Ubuntu 的 Windows 用户还可以选择使用虚拟机安装 Ubuntu，这样你可以在 Windows 下同时操作 Ubuntu，对学习 Ubuntu 有很大的帮助，虚拟机给使用 Ubuntu 的用户提供了一个尝试的平台，你在 Ubuntu 下遇到难题，可以在熟悉的 Windows 中查找资料。而且有些可以使用 Windows 来弥补 Ubuntu 的一些不足。

虚拟机软件可以在一台电脑上模拟出来若干台 PC，每台 PC 可以运行单独的操作系统而互不干扰，可以实现一台电脑“同时”运行几个操作系统，还可以将这几个操作系统连成一

个网络。虚拟机的运行对其他硬件设备要求不是很高，只对内存有一定的要求。一般 512MB 的内存足以运行虚拟机。

本书主要以目前最为流行的虚拟机软件 VMware 来展开。

5.3.1 虚拟机的安装

你可以从网上下载一个 VMware 的软件，可以是最新版的，需要找到相应的注册码，目前网上很容易找到，在这里就不细说了。VMware 的安装也很简单，只要熟悉安装 Windows 软件你就可以成功安装 VMware 软件。

下载 VMware 软件之后，解压安装 VMware 软件，输入注册码，就可以使用。

5.3.2 虚拟环境的配置

运行 VMware 软件，可以看到如图 5-11 所示的登录界面。

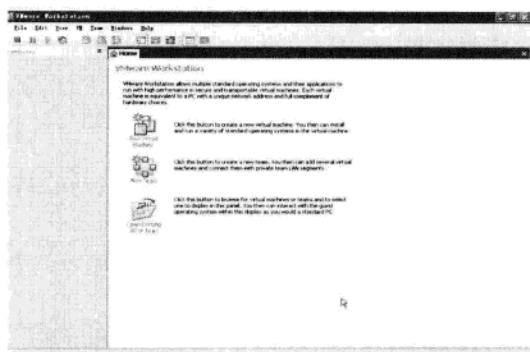


图 5-11 VMware 主界面

首先需要在 VMware 软件中创建一个虚拟机，作为安装 Ubuntu 的虚拟计算机。单击 file | New | Virtual Machine 或是在键盘上按下“Ctrl+N”快捷键，还可以单击界面上的 New Virtual Machine 图标，创建一个新的虚拟机环境。启动创建新虚拟机的引导程序，如图 5-12 所示。

单击“下一步”按钮选择合适的配置，如图 5-13 所示。

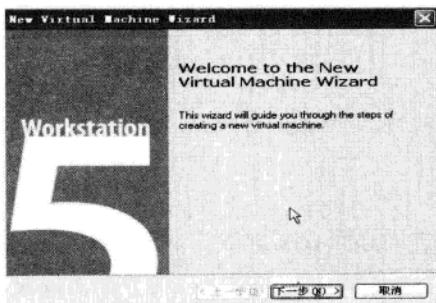


图 5-12 引导创建新虚拟机

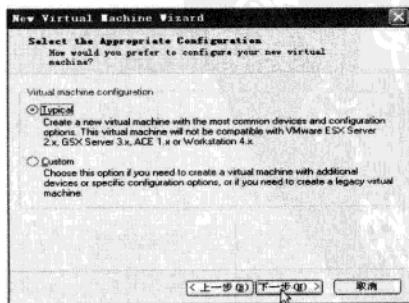


图 5-13 选择配置文件模板



Ubuntu Linux 从初学到精通

这里我们选择 Typical 选项。单击“下一步”按钮，进入下一个界面如图 5-14 所示，选择我们即将需要安装的 Ubuntu 系统。这一步中有两个选项框，第一个为操作系统，第二个为操作系统的版本。在第一选框中有选项：Microsoft Windows, Linux, Novell Netware, Sun Solaris 和 Other。在第二个选框中有相应的操作系统版本。我们在第一选框中选择 Linux，在第二选框中选择 Ubuntu。

单击“下一步”按钮，设置虚拟机的名称和保存的路径，如图 5-15 所示。

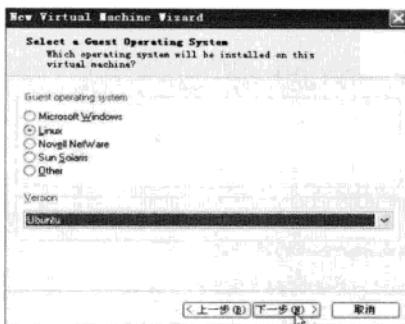


图 5-14 选择需安装的操作系统

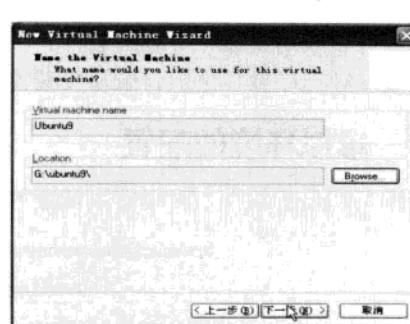


图 5-15 命名虚拟机

在一个足够大的磁盘分区上，创建一个新的文件夹名为 Ubuntu9，设置 location 的值。单击“下一步”按钮，进入选择网络类型界面，如图 5-16 所示。

选择默认的选项就可以。单击“下一步”按钮，指定虚拟磁盘大小，一般推荐的为 8GB 空间，如图 5-17 所示。

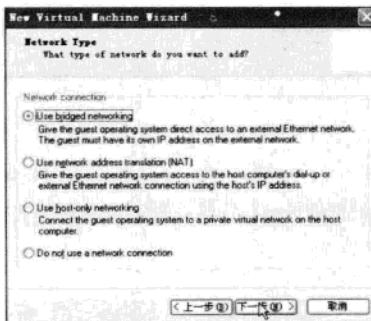


图 5-16 选择网络类型

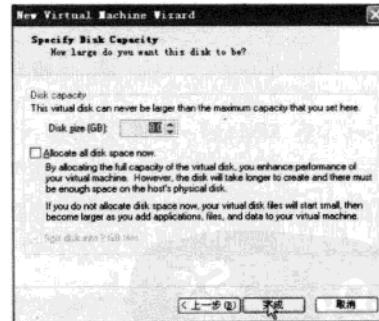


图 5-17 指定磁盘大小

读者可以根据自己的需要修改虚拟磁盘的大小。单击“完成”按钮，完成了虚拟机的创建。

5.3.3 安装过程

我们需要使用映射文件来启动、设置虚拟光盘。如图 5-18 所示。

对于只有 256MB 的虚拟内存，我们觉得还是不够大，也同样可以在界面上修改。把虚拟内存条的容量调到 512MB，调整虚拟内存条的容量，如图 5-19 所示。

单击左边的“Power On”按钮，启动虚拟计算机，如图 5-20 所示。

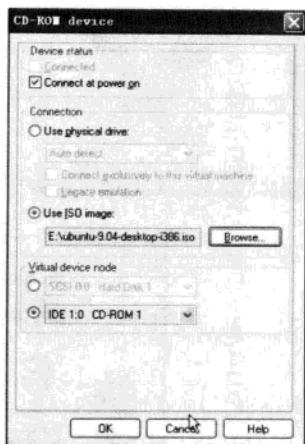


图 5-18 设置映像文件启动

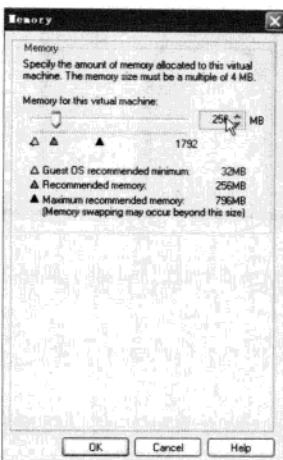


图 5-19 调整虚拟内存条的容量

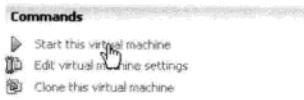


图 5-20 启动虚拟计算机

进入启动界面你可以看到如同计算机的启动界面。由于我们使用了虚拟机技术，那么接下来的安装工作就如同在第 4 章中说的在单机上安装 Ubuntu 的步骤是一样的。这里就详细说明安装步骤。进入 Live CD 系统，对 8GB 的磁盘空间进行分区，单击 Live CD 系统桌面上的“安装”按钮就可以开始安装 Ubuntu。

5.4 小结



本章主要讲述了在不使用光驱的情况下安装 Ubuntu 9.04 的 3 种方法：Wubi 安装方式，GRUB4DOS 安装方式和虚拟机安装方式。

在 Wubi 安装的时候，我们需要注意的是映像文件所在分区的文件系统类型。Wubi 安装简单快捷，不需分区，不怕对磁盘产生损害。

GRUB4DOS 安装 Ubuntu 9.04 拥有比较大的发挥空间，自由度比较大。需要最新版本的 GRUB4DOS 软件的支持。设置好 GRUB4DOS 的启动菜单，就可以轻易地安装系统。

虚拟机安装方式，可以直接在 Windows 平台下操作 Ubuntu 系统，从安装到运行都在 Windows 平台下。虚拟机为 Ubuntu 创建了一个虚拟的计算机，Ubuntu 安装并运行在这一虚拟机上。随时可以通过按下“Ctrl+Alt”组合键从 Ubuntu 平台切换到 Windows 平台。

5.5 习题



- 什么是 Wubi？Wubi 安装 Ubuntu 有什么好处？
- 选择一种适合你的安装方式，并按照上述所说的方法安装 Ubuntu 9.04。
- Wubi 安装需要注意什么？GRUB4DOS 需要注意什么？

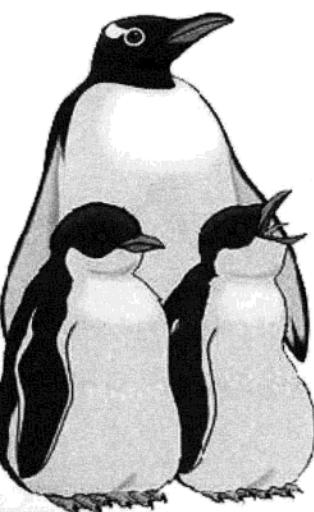
Linux

第6章

快速配置系统

无论通过哪种方式安装 Ubuntu 9.04 系统，默认安装的 Ubuntu 系统都不可能完全满足所有用户的需求，这就要求对刚安装的 Ubuntu Linux 系统进行一番设置。

本章主要讲述一些常用的配置，让读者熟练地使用 Ubuntu，做到上网、娱乐、工作无障碍。这些配置包括：分辨率的设置，屏保的设置，桌面设置，语言配置，输入法配置，软件源的设置和网络连接的配置等。



6.1 语言支持设置



尽管我们安装 Ubuntu 9.04 的时候选择了中文简体，可 Ubuntu 系统开始运行的时候，仍然不能很好地支持中文，如系统中菜单的名称都是中英文混合的。为了使 Ubuntu 系统能够完美地支持中文，用户需要设置系统支持的语言。选择“System（系统）”|“Administration（系统管理）”|“Language Support（语言支持）”菜单命令，启动语言支持工具，如图 6-1 所示。

启动语言支持工具之后，语言支持工具会自动检测系统的语言包是否已完全安装，若系统没有完全安装语言支持，则弹出警告对话框如图 6-2 所示。

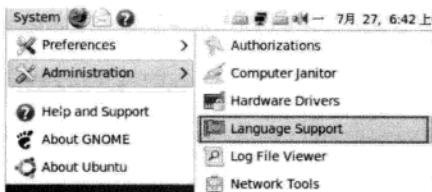


图 6-1 启动语言支持工具

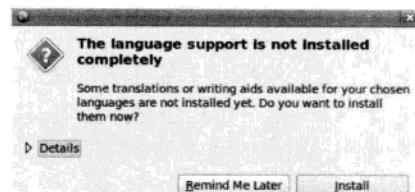


图 6-2 提示语言支持并没有完全安装

用户若想完全安装默认语言支持则可单击“Install”按钮下载并安装默认语言支持的剩余软件包；若只想安装中文支持，那么单击“Remind Me Later”按钮进行选择性安装。

单击“Remind Me Later”按钮，显示 Language 对话框如图 6-3 所示。在选项“For my menus and windows, use”的下拉列表中选择“汉语”选项作为当前登录用户的菜单和窗口显示的文字。在选项“For everyone at startup and login, use”的下拉列表中选择“汉语”选项作为系统中其他用户启动时使用的语言；若选择“Keep the same”选项，则无须在下拉列表中作出选择，它表示其他用户和当前用户使用相同的语言。

尽管设置了这些选项，Ubuntu 系统的菜单和窗口上的标题并没有全部显示为中文，是因为系统还没有完全支持中文。这就要求安装 Ubuntu 的中文支持软件包。单击“Install/Remove Languages...”按钮，弹出 Installed Languages 对话框如图 6-4 所示。

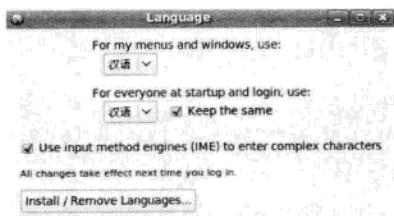


图 6-3 Language 对话框

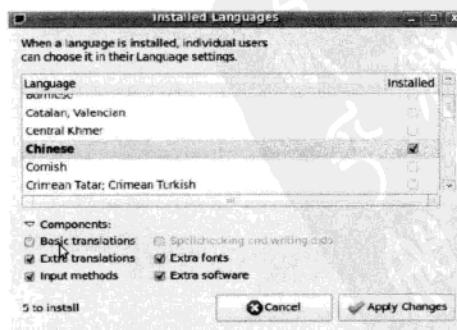


图 6-4 安装语言支持



在语言列表中选中 Chinese，并选中“Installed”复选框，然后单击“Apply Changes”按钮。系统弹出提示框，要求用户输入用户密码，如图 6-5 所示。在 Password 输入框中输入用户密码，单击“OK”按钮，弹出 Downloading Package Files 对话框，如图 6-6 所示，开始下载中文支持所需的软件包。

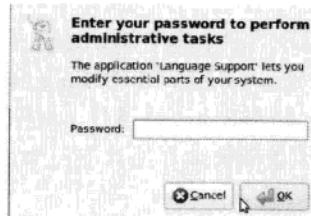


图 6-5 输入用户密码

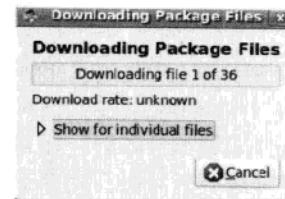


图 6-6 下载所需文件

支持中文的软件包下载完毕之后，系统会自动安装，安装完毕弹出 Changes applied 对话框，单击“Details”下拉按钮显示安装支持软件包后的执行结果，如图 6-7 所示。

单击“Close”按钮关闭对话框，完成 Ubuntu 的汉化。选择如图 6-8 所示的“dranyu”|“Log Out”菜单命令，注销当前用户。系统重新启动之后，Ubuntu 就能完美地支持简体中文，它的登录界面，桌面系统和菜单等的字体都显示为简体中文。

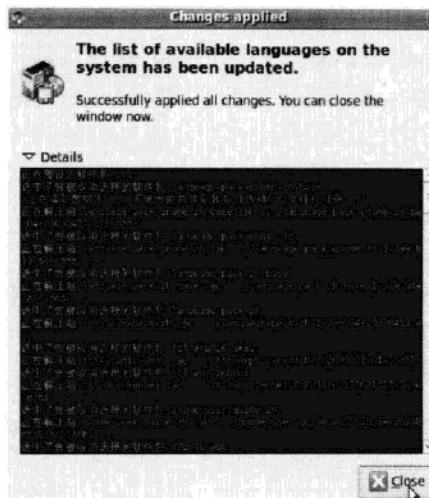


图 6-7 Changes applied 对话框

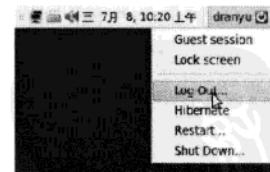


图 6-8 注销当前用户

6.2 设置 SCIM

SCIM 是 Ubuntu 默认安装的一个输入法程序。SCIM 程序向用户提供多种语言的输入法。默认的情况下，只有英文输入。完成语言支持之后，重启计算机，此时中文的输入法插件已



添加到 SCIM 程序中。选择“系统”|“首选项”|“SCIM 输入法设置”菜单命令，弹出 SCIM 输入法设置对话框，如图 6-9 所示。

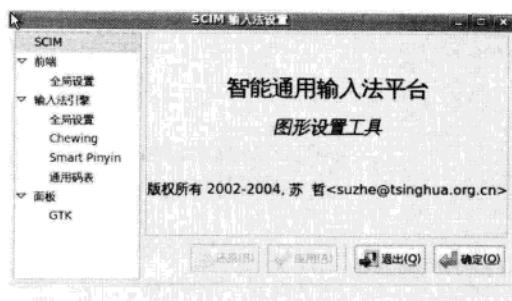


图 6-9 SCIM 输入法设置

SCIM 输入法设置拥有 3 个设置模块：前端设置模块，输入法引擎设置模块和面板设置模块。前端设置模块主要设置 SCIM 输入作为工具应用于其他程序的方式，以及设置 SCIM 操作的快捷方式。输入法引擎设置模块主要设置已安装到 SCIM 中的输入法引擎输入字符的方式和激活状态，面板设置模块主要设置面板的 SCIM 工具条的显示方式，输入窗口中选词方式。

6.2.1 设置 SCIM 开关键热键

设置左边“Ctrl+Shift”组合键为开关键步骤如下：

(1) 单击 SCIM 启动界面左边的“前端”下拉按钮，选择“全局设置”选项，全局设置界面如图 6-10 所示。SCIM 默认使用“Ctrl+Space”组合键作为 SCIM 的开关键。



图 6-10 前端设置模块

(2) 在热键选项框中，单击开关键选项中的设置按钮 ，弹出选择输入法开关键对话框如图 6-11 所示。从图中可以发现在选中的按键列表中显示了已定义的开关键。

(3) 单击键码栏后面的设置按钮 ，弹出捕捉按键对话框如图 6-12 所示。

(4) 此时，在键盘上按下键盘左边的“Ctrl+Shift”组合键，而后释放按键，捕捉按钮对话框消失，返回到选择输入法开关键对话框，用户按下的组合键显示在键码和修饰键选项上。



单击“添加”按钮，将组合键添加到选中的按键列表中，新添加的组合按键显示为 Control+Shift+ Shift_L +KeyRelease，如图 6-13 所示。



图 6-11 选择输入法开关键

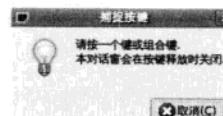


图 6-12 捕捉按键



图 6-13 单击添加按钮后

(5) 单击“确定”按钮，完成添加左边“Ctrl+Shift”组合键，回到前端设置模块的全局设置对话框，单击“应用”按钮，将设置应用到 SCIM 输入法中。此时用户可按下键盘左边的“Ctrl+Shift”组合键进行开启和关闭 SCIM 输入法工具条的操作。SCIM 输入法默认的“Ctrl+Space”开关键依然有效，由于我们并没有把这一组合键删除，SCIM 允许存在多个组合键执行相同的操作。

6.2.2 禁止部分输入法引擎

SCIM 输入法程序默认激活所有添加到 SCIM 的输入法引擎，但并不是所有输入法引擎都需要被使用。这样为了简化 SCIM，我们可以使一部分输入法引擎失效，激活所需的输入法引擎，设置步骤如下：

(1) 单击 SCIM 启动界面左边的“输入法引擎”下拉按钮，选择“全局设置”选项，全局设置界面如图 6-14 所示。



图 6-14 输入法引擎设置

(2) 在“已安装的输入服务”列表中，显示了目前 SCIM 已安装的输入法引擎。单击展开 \triangleright 按钮，显示各具体的输入法引擎。单击激活列的可选按钮，启动和取消激活相应的输入引擎，操作效果如图 6-15 所示。

(3) 单击“确定”按钮，完成设置。单击 GNOME 桌面顶部面板上的输入法图标，显

示已激活的输入引擎如图 6-16 所示。从菜单中选择“简体中文-五笔字型”选项，就可切换至五笔输入法引擎，并且使用五笔输入法添加中文文字。

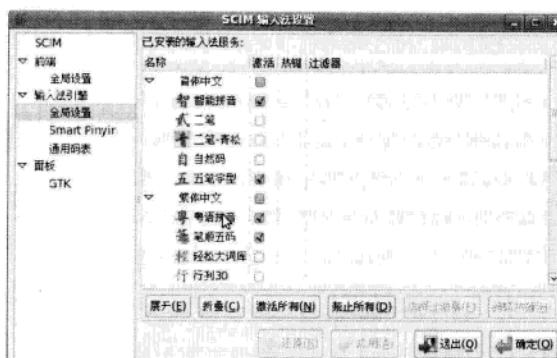


图 6-15 显示输入法引擎

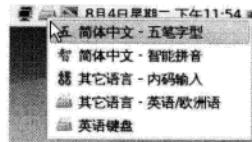


图 6-16 已激活的输入法引擎

6.2.3 设置垂直式候选词表

按下“Ctrl+Space”组合键，就可显示 SCIM 的工具条，如图 6-17 所示。单击工具条上的图标，可以从下拉菜单中选择所需的输入法。当在应用程序的输入框中输入文字时，SCIM 的输入窗口会自动弹出。在默认情况下，输入窗口为水平式候选词表，如图 6-18 所示。设置输入窗口为垂直式候选词表的步骤如下：

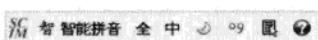


图 6-17 SCIM 工具条

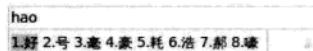


图 6-18 SCIM 输入窗口

- 单击 SCIM 启动界面左边的“面板”下拉按钮，选择“GTK”菜单，GTK 界面设置如图 6-19 所示。
- 在输入窗口选项框中选择“垂直式候选词表”选项，单击“确定”按钮后，在输入框中输入文字时，就会出现垂直式候选词表的输入窗口，如图 6-20 所示。

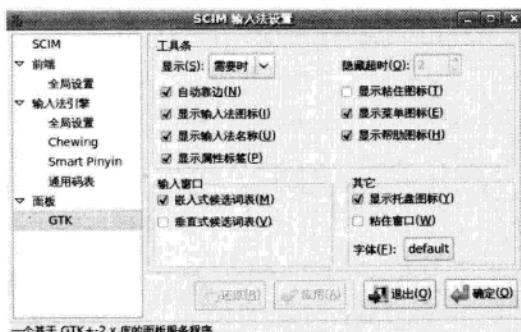


图 6-19 GTK 面板设置



图 6-20 垂直式候选词表



6.3 设置日期时间

安装系统的第二步是选择时区，系统的时间就已被设置。不过在安装过程中，一些用户没设置好合适的时区，在安装之后用户同样可修改系统时间。此修改系统日期时间的步骤如下：

(1) 选择“系统”|“系统管理”|“时间和日期”菜单命令，弹出时间和日期设置对话框，如图 6-21 所示。此对话框呈灰色，表示此对话框不可编辑，解锁后可被编辑。

(2) 单击“解锁”按钮，弹出认证对话框，如图 6-22 所示。

(3) 在“dranyu 的密码 (P)”输入栏中输入用户的密码，然后单击“认证”按钮。认证成功后，返回到时间和日期设置对话框，此时对话框已被解锁。

(4) 在时间选项中，在第一个输入框输入小时数字 23，在第二个输入框输入分钟数字 25，在第三个输入框输入秒钟数字 40。在日期选项框中，单击日历列表中的数字 9。操作结果如图 6-23 所示。



图 6-21 时间和日期设置

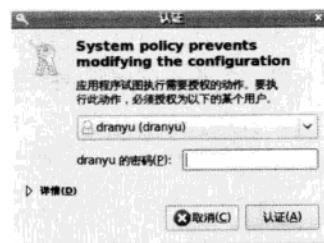


图 6-22 认证对话框

单击“关闭”按钮，关闭时间和日期设置对话框，完成系统的日期和时间设置。系统的
时间会显示在桌面系统的顶部面板上，如图 6-24 所示。



图 6-23 已解锁



图 6-24 系统时间显示



6.4 设置屏幕保护

Windows 用户都知道 Windows 操作有屏幕保护程序，同样 Ubuntu Linux 也有自己的屏幕保护程序，它提供了多种绚丽的屏保程序。同时 Ubuntu 还允许用户添加自己喜欢的图片作为启动屏幕保护程序播放的图片。设置屏幕保护程序播放指定图片文件的具体步骤如下：

(1) 选择“系统”|“首选项”|“屏幕保护程序”菜单命令，弹出屏幕保护程序首选项对话框，在屏幕保护程序主题列表中选中“图片文件夹”选项，如图 6-25 所示。单击“预览”按钮可预览屏幕保护程序启动后的效果。此处“图片文件夹”指向的是用户主目录下的图片文件夹 (/home/用户名/图片)。

屏幕保护程序主题列表中显示了 Ubuntu 提供的多种屏幕保护的主题程序。在“于此时间后视计算机为空闲”选项的滑动栏，10 分钟表示计算机不操作 10 分钟后启动屏幕保护程序。选中“屏幕保护程序激活时锁定屏幕”选项，表示用户从屏保回到系统时需要输入用户密码。



图 6-25 屏幕保护程序首选项

(2) 单击“关闭”按钮，退出屏幕保护程序选项对话框。当计算机 10 分钟不操作时，屏幕保护程序就会启动并且播放图片文件夹里面的图片。

6.5 设置外观



选择“系统”|“首选项”|“外观”菜单命令，弹出外观首选项对话框，如图 6-26 所示。在此对话框中，用户可设置 GNOME 桌面的主题、桌面的背景、系统使用过程中各应用程序的默认字体、菜单和工具条的显示方式，以及桌面的三维显示效果。



6.5.1 修改主题

在外观首选项对话框中，选择“主题”选项卡，然后在主题列表框中选择一款自己喜欢的主题——清爽，则清爽主题即刻应用于 GNOME 桌面系统中，如图 6-27 所示。



图 6-26 外观首选项主界面



图 6-27 设置清爽主题

6.5.2 自定义主题

用户除了更换 Ubuntu 已定义的默认主题之外，还可自定义主题。自定义主题的具体设置步骤如下：

(1) 在外观首选项对话框中，选择“主题”选项卡，然后单击“自定义”按钮，弹出自定义主题对话框如图 6-28 所示。

(2) 在自定义主题对话框中，用户按自己的要求设置完毕之后，单击“关闭”按钮，关闭自定义主题对话框，回到外观首选项对话框，在主题列表框中显示已添加了一个名为“自定义”的新主题，如图 6-29 对话框。



图 6-28 自定义主题



图 6-29 添加了“自定义”主题

(3) 选中刚创建的新主题即“自定义”主题，单击“另存为”按钮，弹出“主题另存为...”

对话框，如图 6-30 所示。

(4) 在名称输入框中输入“mine”，在描述文本框中输入对此主题的描述，单击“保存”按钮，退出此对话框。可看到在主题列表框中创建了一个名为 mine 的主题，如图 6-31 所示。



图 6-30 主题另存为



图 6-31 自定义主题——mine

(5) 单击“关闭”按钮完成自定义主题。

6.5.3 安装新主题

主题的文档必须为 tar.gz 文件。在安装之前必须从网上下载 Moomex 的主题软件包 Moomex.tar.gz，并将软件包保存到本地磁盘。下载 Moomex 主题软件包的网址为：<http://GNO-ME-look.org/CONTENT/content-files/57063-Moomex.tar.gz>。

做好安装准备之后，安装主题包的具体步骤如下：

(1) 在外观首选项对话框中，选择“主题”选项卡，然后单击“安装”按钮，弹出选择主题对话框，如图 6-32 所示。

(2) 在本地磁盘中找到刚下载的 Moomex.tar.gz 主题软件包，选中它，然后单击“打开”按钮，系统将引导安装 Moomex 主题。Moomex 主题安装完毕之后，在主题列表框中显示名为 Moomex 的主题，如图 6-33 所示。



图 6-32 选择主题



图 6-33 Moomex 主题安装成功



(3) 单击“关闭”按钮，关闭外观首选项对话框，完成安装新主题操作。

6.5.4 设置桌面背景

在 Windows 下，用户可以从网上下载美丽的壁纸，并将壁纸设置为系统桌面背景，在 Ubuntu Linux 下，用户同样可以这样做。设置桌面背景的具体步骤如下：

(1) 在外观首选项对话框中，选择“背景”选项卡，显示如图 6-34 所示的背景设置界面。或者在桌面用鼠标右键单击，在弹出的快捷菜单中选择“更改桌面背景”菜单命令，同样打开背景设置界面。其实在桌面背景列表框中有 Ubuntu 提供的默认壁纸，其中第一张是为了设置纯色桌面背景而设计的。

(2) 单击“添加”按钮，弹出添加壁纸对话框，如图 6-35 所示。

(3) 选择所需添加的壁纸，然后单击“打开”按钮，将壁纸添加到桌面背景列表中如图 6-36 所示。

(4) 选中新添加的壁纸，单击“关闭”按钮，完成桌面背景的设置。此时系统的桌面背景已修改为指定的壁纸。



图 6-34 背景设置界面

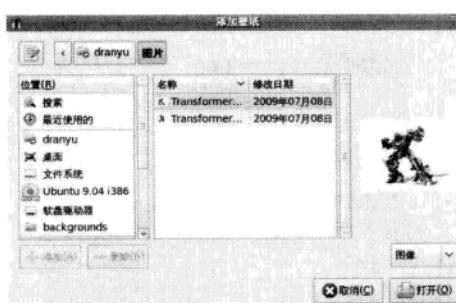


图 6-35 “添加壁纸”对话框



图 6-36 添加壁纸

6.5.5 开启三维显示效果

在外观首选项对话框中，选择“视觉效果”选项卡，显示如图 6-37 所示界面。

图中 3 个选项解释为“无(N)”选项表示没有开启特效功能；“正常(O)”选项的视觉效果提供了提升的可用性，以及介于极具吸引力和中度性能需求的良好平衡；“扩展(X)”选项的视觉效果功能，提供了更多美轮美奂的效果设置，但需要更快的显卡支持。选中 3 个选项之一，系统会自动启动相应选项的效果功能，若用户的显卡不支持此选项，系统会提示不能开启效果功能。

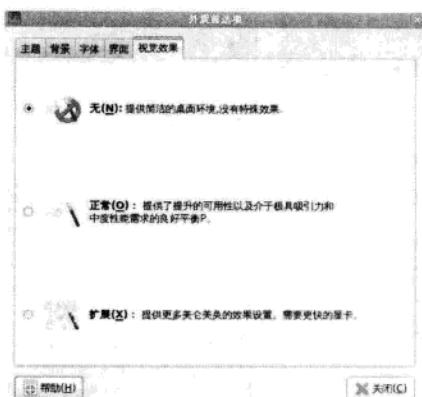


图 6-37 视觉效果设置

6.6 设置分辨率



为了使显示器能够显示更好的图片，我们需要设置显示的分辨率。具体的设置方法为：

(1) 效果选择“系统”|“首选项”|“显示”菜单命令，弹出“显示首选项”对话框如图 6-38 所示。

(2) 打开分辨率下拉列表，在显示如图 6-39 所示的下拉列表中选择“1024×768(4:3)”选项，设置显示器的分辨率。选择“在面板中显示‘显示属性’”选项，首选项的设置图标就会显示在顶部面板上，如图 6-40 所示，单击该图标可快捷地打开显示首选项对话框。

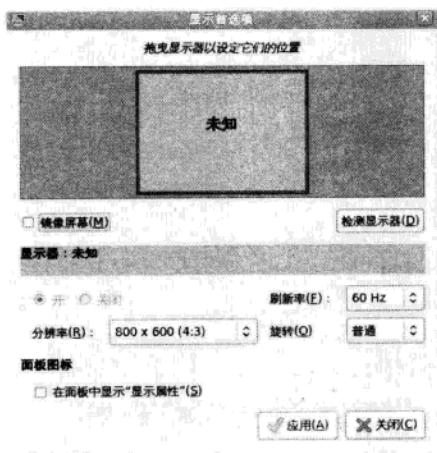


图 6-38 显示首选项

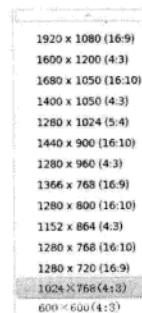


图 6-39 所支持的分辨率

7月28日星期二 上午 9:06



6.7 设置登录窗口

Windows 没有直接提供一个可设置的系统登录窗口给用户，用户需要通过添加其他的软件才可以做到，然而在 Ubuntu 中，用户可以设置本地登录窗口、远程登录窗口，还可以随机选择登录窗口，使得每次登录的登录窗口有所不同，还可设置自动登录，免去了每次登录都要输入账号和密码的麻烦。以下从设置本地登录窗口和设置用户自动登录讲解登录窗口首选项的应用。

6.7.1 设置本地登录窗口

设置本地登录窗口可改变本地登录窗口的外观，具体的设置步骤如下：

(1) 选择“系统”|“系统管理”|“登录窗口”菜单命令，弹出登录窗口首选项设置窗口如图 6-41 所示。

(2) 选择“本地”选项卡，切换到本地登录窗口设置界面，选择样式下拉菜单，选择“带头像浏览器的主题默认”选项；在主题下拉列表中，选择“仅用选中”选项；在主题列表框中选择“Human List”主题选项；在欢迎信息选项框中，选择“自定义”选项，并在输入框中输入“欢迎来到 Ubuntu 的世界”，操作结果如图 6-42 所示。

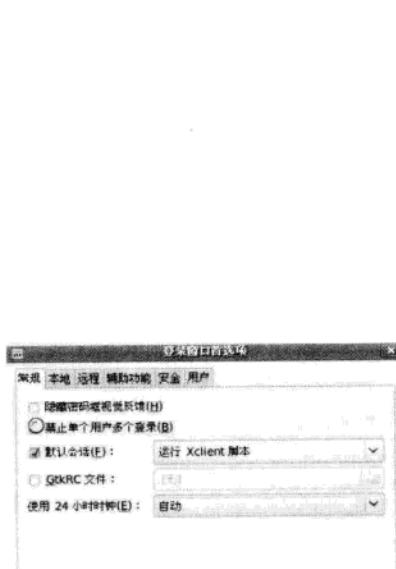


图 6-41 登录窗口首选项



图 6-42 设置 Human List 主题登录窗口

(3) 单击“关闭”按钮，关闭登录窗口首选项，完成设置 Human List 主题的登录窗口操作。退出 Ubuntu 系统后，Ubuntu 的登录窗口如图 6-43 所示。

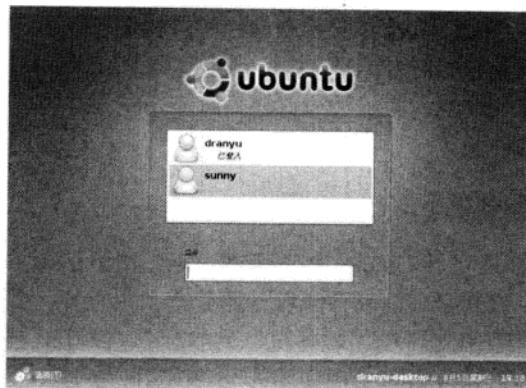


图 6-43 Ubuntu 登录窗口

6.7.2 设置自动登录

设置自动登录可省去登录系统时输入的麻烦，用户启动计算机后，即会自动登录到系统，具体设置步骤如下：

- (1) 选择“系统”|“系统管理”|“登录窗口”菜单命令，弹出登录窗口首选项对话框如图 6-41 所示。
- (2) 选择“安全”选项卡，切换到安全设置界面。在安全设置界面选择“用户自动登录”选项，并在其后的下拉列表中选择“sunny”选项，其他设置不变，如图 6-44 所示。

- (3) 单击“关闭”按钮，完成自动登录的设置。

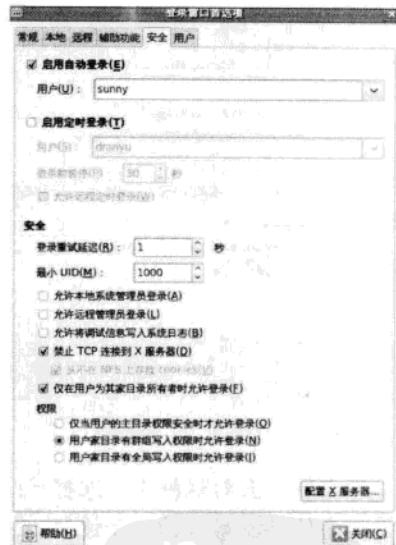


图 6-44 设置自动登录



6.8 设置软件源

Ubuntu的一大特色就是频繁更新软件包，保证用户当前使用的软件包是最新的稳定软件包。这么频繁的更新率，要是没有一个自动更新的机制是难以让用户接受的。Ubuntu 提供了自动更新机制，并且有软件包需要更新的时候，会自动提醒用户执行更新操作。Ubuntu 的软件源提供有多种方式：官方的软件源、第三方的软件源、光盘软件源。安装时，系统会自动配置软件源的位置，如我们把时区设置为中国的上海，那么系统就会把服务器指定为中国附近的服务器，并将相关的信息保存到/etc/apt/sources.list 文件里。



为快速下载更新的软件包，以及在安装软件包时可快速下载所需的安装软件包，设置软件源就变得十分重要。设置软件源有两种方式，一种是通过编辑/etc/apt/sources.list文件，另一种是通过图形界面设置软件源。第一种方式在后续章节讲述，本节中讲述如何使用系统附带的软件源，设置程序软件源。

6.8.1 设置软件源服务器

Ubuntu 在许多国家都设置了 Ubuntu 的服务器，若从附近的服务器下载软件包，下载软件包的时间会大大减少。设置合理的软件源服务器就变得很重要。其设置的具体步骤为：

(1) 选择“系统”|“系统管理”|“软件源”菜单命令，弹出认证窗口，输入用户密码之后单击“确定”按钮，弹出软件源对话框如图 6-45 所示。

(2) 在“下载自”的下拉菜单中选择“其他...”选项，弹出选择下载服务器窗口如图 6-46 所示。

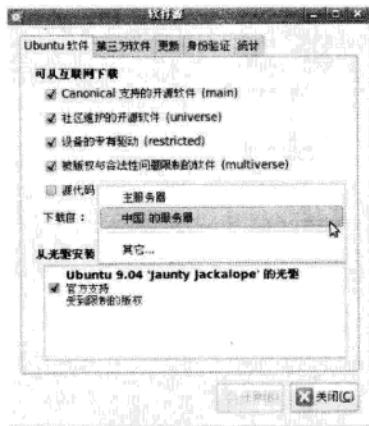


图 6-45 软件源

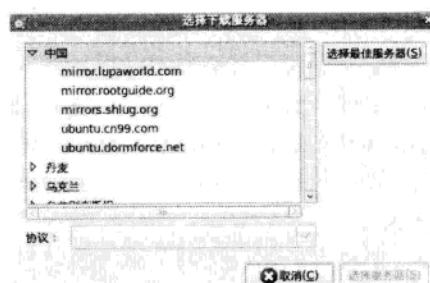


图 6-46 选择下载服务器

在软件源对话框中，选择“Ubuntu 9.04 ‘Jaunty Jackalope’ 的光驱”选项，表示使用光盘作为软件源来源的一个途径。Ubuntu 9.04 的发布光盘中包括了大部分 Ubuntu 使用软件包和一个依赖关系软件包。

(3) 用户选择软件源服务器列表中“中国”菜单下的子菜单“ubuntu.cn99.com”作为服务器。单击“选择服务器”按钮，关闭窗口，返回软件源对话框。或者用户通过单击“选择最佳服务器”按钮，让系统自动选择一个最佳的服务器。

(4) 单击“关闭”按钮，关闭软件源对话框，完成设置。

6.8.2 设置软件更新

在软件源对话框中选择“更新”选项卡，切换到更新设置对话框如图 6-47 所示。在“自动更新”下拉列表中，选择“每天”选项，如图 6-48 所示。

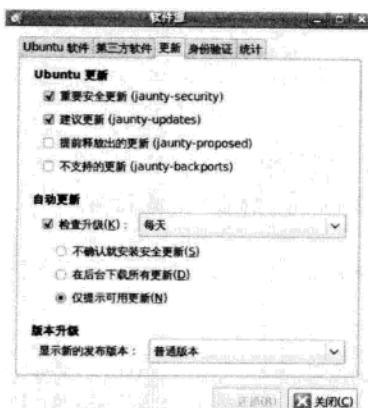


图 6-47 更新设置对话框

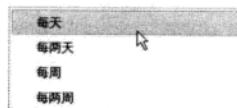


图 6-48 更新的频率选择

设置必要的选项之后，单击“关闭”按钮，关闭软件源对话框，完成设置。

6.9 启动程序



Ubuntu 中的启动程序设置就如同 Windows 的“程序”|“启动”中的应用程序设置一样。在 Ubuntu 的启动程序中，用户可以设置系统运行时需要随着启动的程序，设置的具体步骤如下：

- (1) 选择“系统”|“系统管理”|“启动程序”菜单命令，进入启动程序喜好界面如图 6-49 所示。在“额外的启动程序”列表框中显示了 Ubuntu 默认启动的应用程序。通过单击每个项目左边的可选符号，可激活或禁止开机时启动项目。
- (2) 单击“添加”按钮，弹出添加启动程序界面，如图 6-50 所示。
- (3) 在名称输入框中输入启动程序的名称“Pidgin 通信”；单击“浏览...”按钮，在/usr/bin 目录下选择可执行的应用程序 Pidgin；输入注释“互联网通信程序”。操作结果如图 6-51 所示。

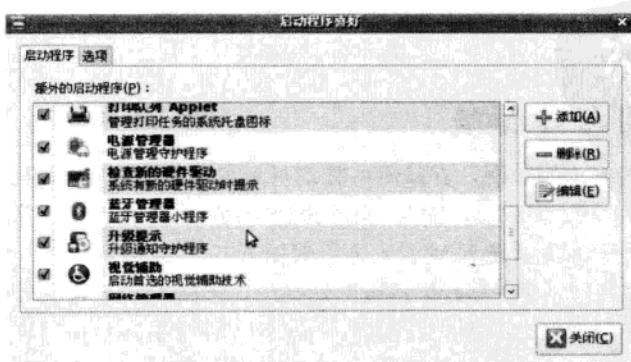


图 6-49 启动程序设置



图 6-50 添加启动程序



图 6-51 设置启动程序

(4) 单击“添加”按钮，完成了启动程序“Pidgin 通信”的添加操作。退出系统，然后重新登录系统，Pidgin 程序就会自动运行。

6.10 网络链接



网络链接是一个比较重要的配置。当前是网络的时代，没有网络，计算机就不能发挥它应有的作用。若用户所在的网络提供的网络链接方式是自动获取 IP 地址方式，那么在用户安装 Ubuntu 之后，Ubuntu 系统会自动获取到网络服务中心动态分配的网络 IP 地址。要是用户通过其他网络链接方式来链接网络的话，那就要对网络链接做相应的修改。Ubuntu 提供多种网络链接配置。本节将讲述配置静态 IP 地址和 ADSL 网络链接方式。

无论设置何种网络链接都需要先打开网络链接对话框，方法为：选择 Ubuntu 顶部面板上的“系统”|“系统管理”|“网络连接”菜单命令，打开网络链接对话框如图 6-52 所示。

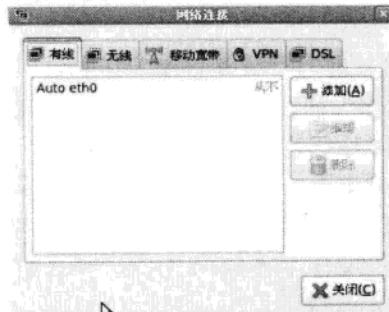


图 6-52 网络链接

6.10.1 设置静态 IP 地址

若用户所在的网络为局域网，并从网络管理员那里获得了网关 IP 地址和 DNS 服务器地址。假设网关地址为 192.168.1.1，DNS 服务器地址为 202.116.138.1，分配的 IP 地址为：192.168.1.166，那么设置静态 IP 地址的具体步骤如下：

(1) 打开网络链接对话框，在网络链接对话框中，选择“有线”选项卡，双击“Auto eth0”记录行或单击“编辑”按钮，弹出正在编辑 Auto eth0 对话框如图 6-53 所示。

(2) 通过正在编辑 Auto eth0 对话框，用户可查看到网卡的信息和修改网络 IP 的获取方式。选择“IPv4 设置”标签，切换到 IPv4 设置界面如图 6-54 所示。



图 6-53 编辑 Auto eth0



图 6-54 设置 IP 获取方式

(3) 在方法的下拉列表中选择“手动”选项，激活其他选项框。单击地址框上的“添加”按钮，则在地址列表框中添加一行空白记录，在“地址”列中输入 192.168.1.166；在“子网掩码”列中输入 255.255.255.0；在“网关”列中输入 192.168.1.1。地址列输入的是用户指定本计算机的 IP 地址，此地址不可以与局域网中其他计算机的 IP 地址重复；网关列输入的是服务器的地址。在 DNS 服务器栏中输入 202.116.128.1，作为 DNS 解析服务器的地址，此地址需要询问局域网的管理员。设置完毕后结果显示如图 6-55 所示。

(4) 若用户所在网络的网络管理员提供了路由的设置，用户可以单击“路由”按钮进行路由设置。

(5) 单击“应用”按钮，关闭正在编辑 Auto eth0 对话框，将配置作用于系统，完成静态分配 IP 地址操作。

6.10.2 设置 ADSL 网络链接

配置静态 IP 地址，是考虑到用户所在的网络为局域网，若用户所在的网络中没有局域网，而是采用 ADSL 网络链接方式上网，那么就需要配置 ADSL 拨号网络链接。用户应该从网络服务提供商处获得 ADSL 网络链接的相关信息，使用这些信息就可设置 ADSL 的网络链接，具体步骤如下：

(1) 设置的开始界面如为有线网络链接和无线网络链接的界面，单击“添加”按钮，弹出编辑 DSL 链接 1 对话框，如图 6-56 所示。

(2) 用户只需设置“DSL”标签上的选项即可。在用户名输入框中输入 DSL 的用户名，在服务输入框中输入服务提供商的服务器名称，在密码输入框中输入服务提供商提供的用户密码，并且选择“自动链接”选项。



图 6-55 设置 IP 地址



(3) 单击“应用”按钮，完成 ADSL 网络链接的设置。

ADSL 网络链接设置完毕，用户可以通过单击顶部面板上的网络链接图标切换网络链接方式。单击网络链接图标弹出下拉菜单如图 6-57 所示。发现菜单中已显示了新添加的“DSL 链接 1”的菜单项，选中此复选框，用户就可以应用 ADSL 网络链接的链接方式了。

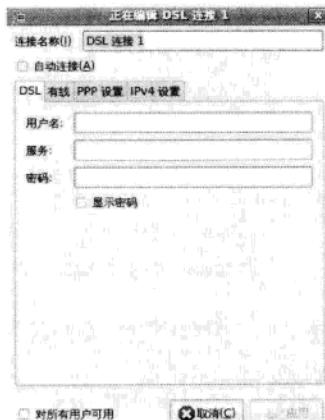


图 6-56 DSL 连接设置



图 6-57 网络链接菜单

6.11 小结



Ubuntu 中针对不同的系统配置提供相关的图形界面配置程序，简化了用户的操作。通过对本章的学习，用户可自行配置个性化的 Ubuntu 操作界面，设置网络链接，设置网络软件源，为自己的工作学习和娱乐提供操作方便而友好的平台。

6.12 习题



1. 从 SCIM 中禁止不经常使用的输入法引擎。
2. 修改屏幕保护程序。
3. 禁止使用的输入法引擎。
4. 将显示器的分辨率设置为 $1024 \times 768(4:3)$ 。
5. 下载一张自己喜欢的壁纸，并将它设置为自己的桌面。
6. 从网上下载一个主题包，并把它安装到外观的主题列表中。
7. 设置显示器的分辨率，以及桌面的背景图片。
8. 假设在一个局域网上，192.168.1.1 为网关，202.116.138.2 为 DNS 服务器，那要如何设置静态 IP 地址？

Linux

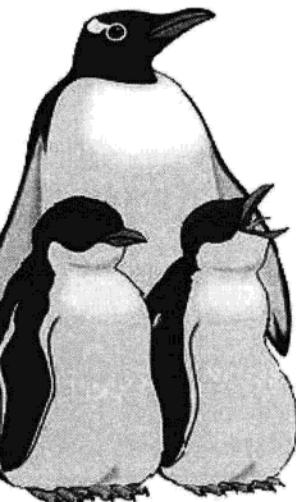
第 3 部分 Ubuntu 的桌面环境

第 7 章

GNOME 桌面环境

桌面环境是指为计算机提供一个图形用户操作界面，典型的桌面环境必须包括提供图标、视窗、工具栏、文件夹、壁纸，以及像拖放这样的能力。Linux 中比较流行的桌面环境为 GNOME 和 KDE，然而在 Ubuntu 系列中使用了 GNOME、KDE 和 Xfce 等桌面环境。Ubuntu 发行版本使用 GNOME 作为默认的桌面环境，Kubuntu 发行版本使用 KDE 作为默认的桌面环境，以及 Xubuntu 使用 Xfce 作为默认的桌面环境。在接下来的几章中，我们将分别介绍这 3 个桌面系统。

本章将介绍 GNOME 的相关知识，描述如何操作 GNOME 的面板和桌面，描述 GNOME 中常用首选项的设置。本章的后面还将简单介绍 GNOME 的常用应用软件。





7.1 GNOME 的介绍

GNOME 是 GNU Network Object Model Environment (GNU 网络对象模型环境英文) 的缩写，是 GNU 计划的一部分，是开放源码运行的一个重要组成部分。其目标是基于自由软件，为 UNIX 或者类 UNIX 操作系统构造一个功能完善、操作简单，以及界面友好的并且是开放自由的桌面环境。

在 GNOME 项目开始之前，KDE 作为一个 Linux 下的桌面系统受广大 Linux 用户所喜爱。但由于 KDE 桌面使用的开发包为 Qt 开发包，而 Qt 开发包是由 Trolltech 公司开发，当时并未使用自由软件许可协议。Qt 所使用软件许可证问题严重制约着 KDE 在 Linux 上的发展。于是在 1997 年，为了解决桌面环境的许可证问题，以及克服 KDE 中开发环境单一 C++ 依赖的问题，Miguel de Icaza 和 Federico Mena 发起了 GNOME 项目。

从一开始，GNOME 就使用了 GNU 自由软件的标准。从而许多热衷于自由软件开发的程序员都参与到 GNOME 项目中，更重要的是一些大公司的加入如 Red Hat Linux。多人参与 GNOME 项目推动着 GNOME 的快速发展。

同时，也是因为 GNOME 使用了 LGPL 协议，GNOME 基金会于 2000 年 8 月份在 Sun、Red Hat 等公司的共同努力下正式成立。GNOME 基金会负责协调 GNOME 的开发和发布，决定哪些部分可以添加到 GNOME 的发布版本中，它并不直接参与 GNOME 项目的技术决策。GNOME 基金会成员的组成有一定的标准，在 GNOME 基金会的网站上有这么一段话描述成员资格的标准定义：

“按照 GNOME 基金会章程，任何对 GNOME 有贡献者都可能是合格的成员。尽管很难精确定义，贡献者一般必须对 GNOME 计划有不小的帮助。其贡献形式包括代码、文档、翻译、计划范围的资源维护，以及其他对 GNOME 计划有意义的重要活动。”

而且 GNOME 基金会成员在每年的 11 月都会选举董事会，并且候选人都必须是对 GNOME 有贡献的人。

在 GNOME 基金会的管理下，GNOME 有一个软件包发展成为一个集成式的桌面环境。目前 GNOME 桌面环境由许多不同的项目构成，其中包括：ATK、GConf、GNOME VFS、GNOME Keyring、GStreamer、GTK+ 和 LibXML 等。GNOME 桌面环境功能在不断地完善，并且一直努力缩小与 KDE 桌面环境之间的差距。目前，许多开放源软件既可以运行在 GNOME 桌面环境下，又可以运行在 KDE 桌面环境下，即为 GNOME 桌面而开发的应用软件可以很好地在 KDE 桌面上运行，相反为 KDE 环境开发的应用软件也可以很好地在 GNOME 桌面上运行。

当前 Ubuntu Linux 9.04 使用了在 2009 年 3 月发布的 GNOME 2.26 版本作为默认的桌面环境。GNOME 2.26 使用最新的 GTK + 2.16 开发包，获得了更佳的图形外观效果，功能也得到了很大的提升。



7.2 GNOME 的面板和桌面

Ubuntu 系统的默认桌面环境为 GNOME 桌面环境。用户登录 Ubuntu 系统之后，进入 GNOME 桌面环境，主界面如图 7-1 所示。

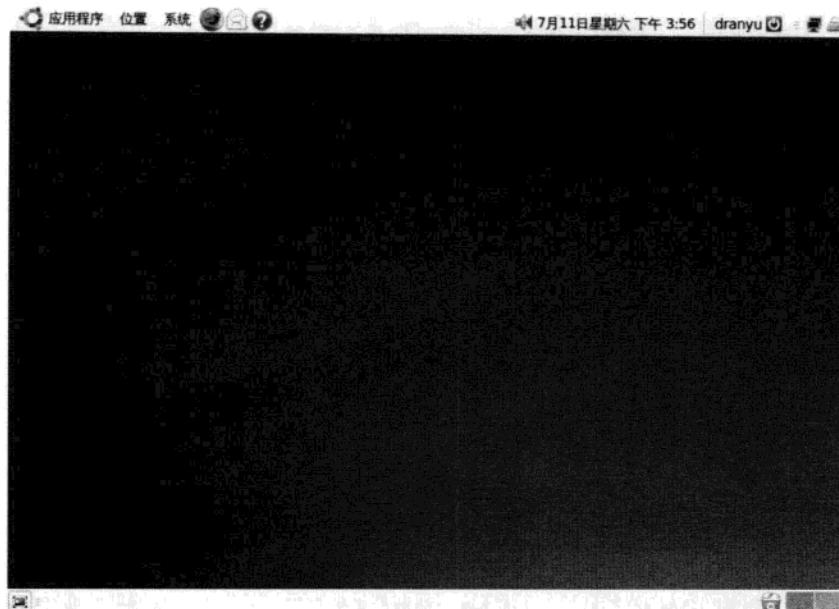


图 7-1 GNOME 主界面

7.2.1 GNOME 面板

图 7-1 中顶部和底部的长条框被称之为面板（Panel）。其中顶部面板默认放置一些常用面板项目如“应用程序”、“位置”、“系统”菜单和快速启动应用软件 Firefox 等。而底部面板放置显示桌面按钮、任务栏、回收站和工作切换器。

1. 顶部面板

默认的顶面板如图 7-2 所示，放置了常用的小应用程序，以及系统的主要菜单。

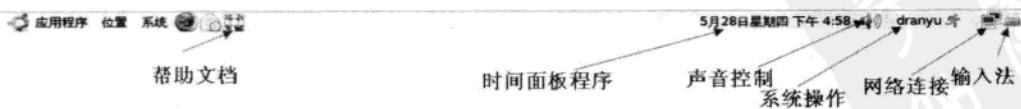


图 7-2 GNOME 顶部面板

- “应用程序”菜单

此菜单下包含了常用应用程序的快速启动菜单。已安装的应用程序大部分都会作为“应



用程序”菜单的子菜单。单击“应用程序”按钮弹出下拉菜单如图 7-3 所示。在应用程序菜单上，单击下一级菜单图标 ，就可以进入到下一级子菜单列表。

- “位置”菜单

位置菜单主要是显示文件系统的目录。用户单击文件目录，可快速进入相应的文件目录，浏览目录里的文件。Ubuntu 还可以连接到网络上其他系统的目录。单击“位置”按钮显示下拉菜单如图 7-4 所示。

- “系统”菜单

此菜单放置了 Ubuntu 系统中的硬件配置和软件配置的可视化配置程序，菜单项列表如图 7-5 所示。

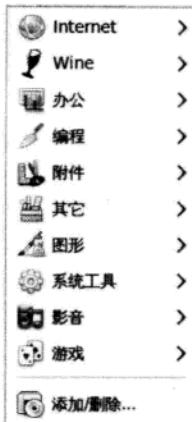


图 7-3 “应用程序”菜单



图 7-4 “位置”菜单

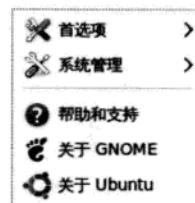


图 7-5 “系统”菜单

“系统”菜单中有两个重要的子菜单“首选项”和“系统管理”。首选项子菜单下放置了系统设置方面的菜单项如图 7-6 所示，而系统管理子菜单放置了系统管理方面的菜单项如图 7-7 所示。

- 系统操作

单击顶部面板上的“系统操作”按钮弹出系统操作菜单如图 7-8 所示。Ubuntu 将系统操作的菜单都放置于此。系统操作菜单中菜单项的含义如下所示。

- “客人会话”: 单击此菜单命令表示不注销当前用户，以 Guest 的账号登录系统。
- “锁住屏幕”: 单击此菜单命令表示不退出系统，锁住屏幕，当用户需要输入密码才能返回系统界面。
- “退出”: 单击此菜单命令表示注销当前用户。
- “休眠”: 单击此菜单命令表示使电脑处于休眠状态，激活时需要输入密码。
- “重启”: 单击此菜单命令表示退出系统并重启计算机。
- “关机”: 单击此菜单命令表示退出系统并关闭计算机。



图 7-6 “首选项”菜单



图 7-7 “系统管理”菜单

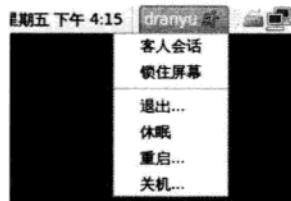


图 7-8 系统操作

2. 底部面板

Ubuntu 另一个默认面板是底部面板，底部面板上的默认项目分布如图 7-9 所示。

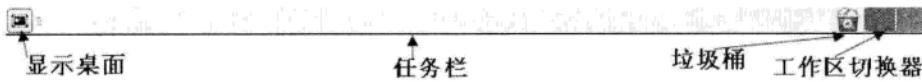


图 7-9 底部面板

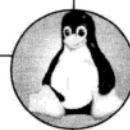
底部面板上各项目的含义如下：

- (1) 单击“显示桌面”按钮可将当前桌面上的所有应用程序窗口最小化。
- (2) 任务栏上显示了已打开应用程序的最小化图标，单击应用程序最小化图标就可以激活相应的应用程序。
- (3) 单击“垃圾桶”图标，就可以进入到垃圾桶文件夹。
- (4) 工作区切换器的主要功能就是管理和切换工作桌面。单击工作区切换器上的矩形图标就可以切换至相应的工作桌面。

7.2.2 面板操作

Ubuntu 允许用户自由设置面板，提供了管理面板的功能，如添加新面板，设置面板的外观，添加新项目等。在本小节中介绍以下面板操作：

- 添加项目到面板



项目指的是可以运行在面板上的小程序。添加项目到面板上，也就是激活小程序并将其放置到面板上。以向顶部面板添加抽屉小程序项目为例讲述添加项目到面板的功能，具体操作步骤如下：

(1) 用鼠标右键单击顶部面板上的空白区域，弹出面板快捷菜单，如图 7-10 所示。

(2) 选择“添加到面板”菜单命令，弹出“添加到面板”对话框，如图 7-11 所示。

(3) 选择“项目列表中的面板程序抽屉”选项，单击“添加”按钮，相应的面板程序就会添加到顶部面板中，显示效果如图 7-12 所示。用户也可以通过双击时钟项目达到相同的效果。项目列表中显示了由 Ubuntu 默认提供的面板上的小应用程序。

(4) 单击“关闭”按钮完成添加项目到面板的操作。

- 添加到面板(A)...
- 属性(P)
- 删除该面板(D)
- 新建面板(N)
- 帮助(H)
- 关于面板(B)

图 7-10 面板快捷菜单



图 7-11 “添加到面板”对话框

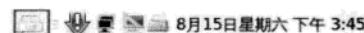


图 7-12 添加抽屉小程序

技术提示

抽屉里边可以放置小程序、启动器等。在 Ubuntu 中使用抽屉可方便快速启动常用的应用程序。向抽屉里添加小程序的方法如同在面板上添加项目的方法。

● 修改面板属性

Ubuntu 允许用户修改面板的属性，使得面板更加符合个人使用的习惯，修改顶部面板属性的具体操作步骤为：

(1) 用鼠标右键单击顶部面板上的空白区域，弹出面板快捷菜单，如图 7-10 所示的面板快捷菜单。

(2) 选择“属性”菜单命令，弹出面板属性对话框，在方向下拉列表中，选择“底部”选项，在大小的数字框中输入 40，选择“自动隐藏”选项，如图 7-13 所示。

(3) 单击“关闭”按钮完成操作。操作结果如图 7-14 所示。

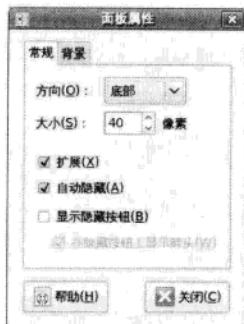


图 7-13 设置面板属性

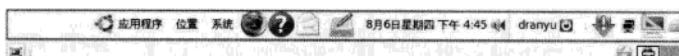


图 7-14 顶部面板属性的调整结果

补充说明

大小表示面板的高度数值。默认面板的大小为 24，最大值可设为 120。选择“自动隐藏”选项，表示当用户的鼠标离开面板一段时间后，面板会自动隐藏到屏幕里去，留下一小部分在屏幕显示区中，当鼠标移动到留在屏幕显示区的那一小部分的面板后，面板就会自动弹出来，使得桌面的可操作范围加大。

- 新建面板

选择图 7-10 上的“新建面板”菜单命令，系统会自动为用户添加一个没有项目的空面板。在新建的空面板上，用户可添加项目，设置新面板的属性。

- 管理面板上的项目

面板上的项目并不是固定的，用户可以移动面板上项目的位置。以时间项目为例简述如何管理面板上的项目。



图 7-15 时间项目快捷菜单

用鼠标右键单击顶部面板的时间项目，弹出时间项目的快捷菜单，如图 7-15 所示。其中“从面板上删除”、“移动”和“锁定到面板”这 3 个菜单，是用鼠标右键单击面板上任何项目都拥有的菜单。“从面板上删除”菜单表示从面板上删除相应的项目。

若选中“锁定到面板”选项，那么“移动”菜单不可用，项目不可被移动。取消“锁定到面板”选项，选择“移动”菜单命令，移动鼠标就可以移动面板上的时间项目。

7.2.3 GNOME 桌面

GNOME 桌面相当简洁，安装 Ubuntu 之后，用户可能只看到 DVD 的图标，这是用户安装时还没有把安装光盘退出的缘故。除此之外，用户看不到其他应用程序的图标。Ubuntu 允许用户添加如同 Windows 下快捷方式的启动器，方便用户启动日常使用的应用程序。以添加启动器 esvn 为例，说明在桌面上添加启动器的方法，具体操作步骤如下：



(1) 单击鼠标右键，弹出快捷菜单如图 7-16 所示。选择“创建启动器”菜单命令，弹出创建启动器对话框如图 7-17 所示。

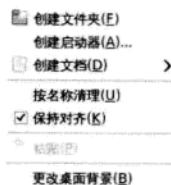


图 7-16 桌面快捷菜单

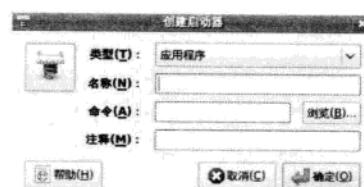


图 7-17 创建启动器

(2) 在类型下拉列表中选择“应用程序”选项。其类型下拉列表中包括三种类型：应用程序、终端中的应用程序和位置。应用程序选项表示即将创建一个图形应用界面的应用程序。终端中的应用程序选项表示即将创建一个运行在终端上的应用程序。而位置选项表示即将创建指向指定文件夹的启动程序。

(3) 在名称栏上输入启动器的名称 esvn。在命令选择框中，单击后边的“浏览”按钮，从文件系统中选取所需执行的应用程序命令 esvn (/usr/bin/esvn)。在注释栏输入“esvn 管理软件源码”，表示此启动器要打开的应用程序 esvn 是管理软件源码的图形化管理工具，设置完毕，显示结果如图 7-18 所示。

(4) 完成设置后，单击“确定”按钮，完成 esvn 启动器的创建。在桌面上就会出现 esvn 启动器的图标，如图 7-19 所示。



图 7-18 设置 esvn 启动器



图 7-19 桌面上显示了 esvn 启动器图标

补充说明

修改 GNOME 桌面背景图片的功能为选择桌面快捷菜单上的“更改桌面背景”命令，然后在弹出的外观首选项对话框中设置桌面的背景，这一部分的设置在第 6 章中已讲述。

7.3 GNOME 首选项



GNOME 桌面提供使用图形界面修改软硬件配置的功能，而且它将这些配置作为菜单放置在系统菜单中。外观首选项、SCIM 设置、屏幕保护程序的设置、时钟设置、网络链接和登录窗口在第 6 章中已讲述，在这里不做详述。下面就几个主要的功能展开讲解。

7.3.1 主菜单

“主菜单”首选项，主要是设置顶部面板上的“应用程序”菜单和“系统”菜单。选择“系统”|“首选项”|“主菜单”菜单命令，弹出主菜单对话框，如图 7-20 所示。



图 7-20 主菜单首选项

在菜单栏上包括“应用程序”和“系统”菜单，它们与 GNOME 面板的顶部面板的“应用程序”和“系统”相对应。单击左边的菜单栏，在中间的项目栏上显示相应的应用程序。若项目记录前端的小方框已被勾选，表示激活此项目，使其在顶部面板上的相应菜单上显示。

技术提示

除了默认的应用程序可以添加到菜单上之外，用户也可以添加已安装的应用程序到菜单中。单击图 7-20 上右边的“新建项目”按钮，打开创建启动器对话框，之后的创建步骤如同 7.2.3 节创建桌面启动器的方法。

7.3.2 首选应用程序

首选应用程序指定打开一个文件的时候，GNOME 桌面应该执行哪个应用程序打开文件？例如用户指定 Mozilla Firefox 网页浏览器作为浏览网页文件的默认应用程序，当桌面上接收到打开网页文件的请求的时候，就会启动 Firefox 并打开相应的网页文件。指定 Firefox 作为默认打开网页文件的应用程序的具体步骤如下：

(1) 选择“系统”|“首选项”|“首选应用程序”菜单命令，弹出首选应用程序对话框，选择“Internet”选项，然后在 Web 浏览器选项框中，从下拉列表中选择 Firefox 选项，选中“在新标签中打开链接”选项，如图 7-21 所示。选中“在新标签中打开链接”选项，表示 GNOME 接收到要求打开页面文件时，GNOME 调用 Firefox 应用程序打开文件，并在 Firefox 浏览器的新标签页中打开页面。



图 7-21 首选应用程序

(2) 单击“关闭”按钮，完成设置 Firefox 为打开网页文件的默认应用程序。

技术提示

除了可设置网页文件的默认浏览器，用户还可指定阅读邮件的默认阅读器程序，默认的视频和音频的播放器，默认终端模拟器等。

7.3.3 键盘快捷键

使用快捷键，可提高工作的效率，减少时间的花费。不同的用户，对快捷键的使用习惯也不同，如有些人在游戏中喜欢使用“Ctrl+Space”组合键作为跳跃，有些用户就习惯了使用“Ctrl+Shift”。Ubuntu 提供给用户更广的自由空间，允许用户自行设置应用快捷键，还可添加自定义快捷键。

GNOME 桌面的快捷键列表中提供了默认的 60 多个快捷键。自定义快捷键的功能更加吸引用户，设置自定义快捷键，可快速打开用户经常使用的应用程序。用户无须使用鼠标就可以完成启动应用程序的操作。假如用户喜欢使用终端应用程序，可设置 Ctrl+1 的组合键作为启动终端应用程序的快捷键方式，具体的设置步骤如下：

(1) 选择“系统”|“首选项”|“键盘快捷键”菜单命令，弹出键盘快捷键对话框，如图 7-22 所示。

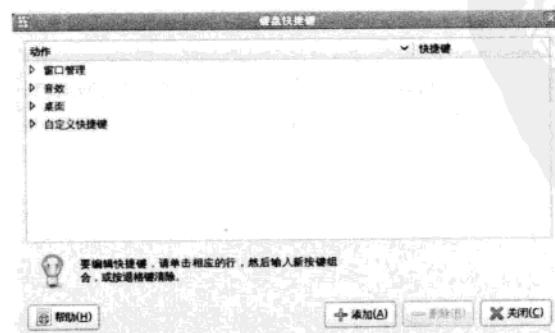


图 7-22 键盘快捷键

(2) 单击“添加”按钮，弹出自定义快捷键对话框如图 7-23 所示。

(3) 在名称栏中输入名称 terminal，在命令输入框中输入需要执行的命令 GNOME-terminal，单击“应用”按钮完成自定义快捷键的定义。输入栏中的命令可以是一个命令，也可以是命令的文件路径 (/usr/bin/GNOME-terminal)。

(4) 在列表中，名为 terminal 的新快捷键就会在“自定义快捷键”分类下显示，此时快捷键列上的值为“已禁止”，用鼠标单击“已禁止”列表框，出现“新建快捷键”字样，并覆盖了“已禁止”如图 7-24 所示。此时，在键盘上按下“Ctrl+1”组合键，“Ctrl+1”组合键就会显示在快捷键列上。若用户不明白操作，可以查看键盘快捷键对话框底部的提示，如图 7-24 底部的操作提示。



图 7-23 自定义快捷键

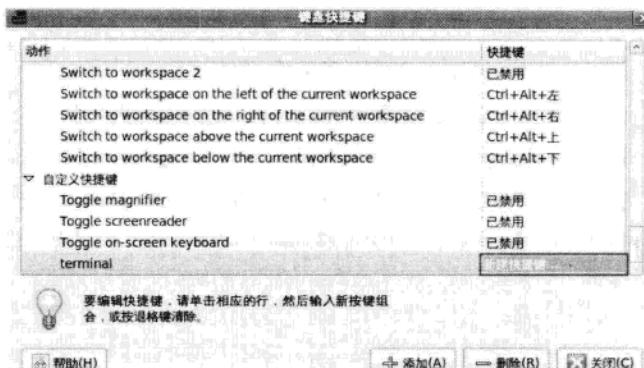


图 7-24 定义快捷键的组合键

(5) 单击“关闭”按钮，关闭键盘快捷键对话框，完成自定义快捷键的添加操作。设置完毕之后，只要用户按下“Ctrl+1”组合键，就可以启动 GNOME-terminal 应用程序。

技术提示

用户在添加自定义快捷键的同时也可以删除自定义快捷键，而且只有自定义的快捷键可以被删除。删除自定义快捷键的方法为：在键盘快捷键对话框中，选中自定义快捷键的记录行，然后单击“删除”按钮，就完成了删除自定义快捷键的操作。

Ubuntu 提供的默认快捷键不可以被删除，但可以被修改，修改默认快捷键的方法为：在键盘快捷键对话框中，选中需要修改的默认快捷键记录行，单击“快捷键”按钮，在键盘上按下需要设置的组合键，即可修改默认快捷键。

7.4 GNOME 应用软件



GNOME 的发布版本中包括了许多常用的应用软件，如时钟工具、文本编辑器、计算器、看图软件等。



7.4.1 字符终端

Ubuntu 默认自带 GNOME 终端——GNOME-terminal 程序。Ubuntu 提供了许多系统应用的命令，它们的运行需要一个 shell 环境支持，GNOME-terminal 就是提供了运行 Ubuntu 命令行的环境。选择“应用程序”|“附件”|“终端”菜单命令，打开 GNOME-terminal 终端程序如图 7-25 所示。用户可以在光标处输入命令，然后按下回车键，就可以在终端上执行命令。如在光标处输入 ls -l 命令，然后按下回车键，执行命令，在命令行的下边会显示当前～目录下的文件或文件夹列表。其中～表示的是用户主目录。



图 7-25 终端

7.4.2 文本编辑器

GNOME 提供了一个功能强大的文本编辑器—Gedit。它能够满足用户文本编辑过程中的一般需求。Gedit 程序使用简单，如同 Windows 的 Notepad。启动 Gedit 应用程序的方式有很多种，其中包括：双击文本文件；选择顶部面板上“应用程序”|“附件”|“文本编辑器”菜单命令；在终端执行 Gedit 命令。Gedit 编辑主界面如图 7-26 所示。

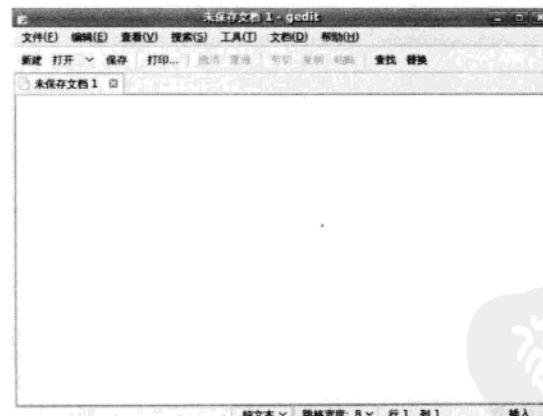


图 7-26 Gedit 文本编辑器

7.4.3 计算器

GNOME 桌面系统还提供了一个简单的计算器程序。用户可根据需要选择基本、高级、财务、科学和编程模式。编程模式可以帮助编程人员进行数字进制之间的转化。在顶部面板上选择“应用程序”|“附件”|“计算器”菜单命令，启动计算器应用程序，打开计算器程序的主界面如图 7-27 所示。



图 7-27 计算器主界面

7.4.4 图像编辑软件

GIMP 是一款功能强大的图像编辑软件。在 Windows 下使用 Photoshop，在 Linux 下就用 GIMP。GIMP 处理软件是搞图形处理的首选编辑软件。从顶部面板上选择“应用程序”|“图形”|“GIMP 图片编辑器”菜单命令，启动 GIMP 图片编辑器，显示主界面如图 7-28 所示。

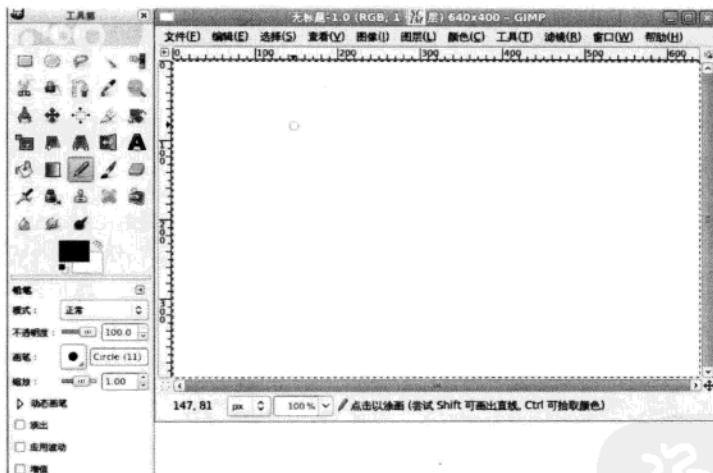


图 7-28 GIMP 主界面

使用 GIMP 图形编辑器，用户可编辑由数码相机拍摄的照片，可绘制图片。在主界面左边的工具栏中显示了许多绘图工具，包括铅笔、画笔、印章等常用的图形处理工具。对于熟悉 Photoshop 的用户很容易就会学会使用 GIMP。

7.4.5 GNOME 媒体播放器

GNOME 中提供了播放音乐、播放视频等多媒体工具。用户选择顶部面板“应用程序”|“多媒体”|“电影播放机”菜单命令，启动应用程序，显示 GNOME 电影播放器主界面如图 7-29 所示。



图 7-29 Totem 电影播放器

电影播放器能播放 DVD，播放来自网络上的电影流文件，还可以安装一些视频的插件观看相应编码的视频。若用户需要观看 YouTube 上的短片，可以在电影播放器中安装一个 YouTube Browser 插件。GNOME 2.26 中电影播放机还可以为视频添加字幕，若电影有多种语言，也可以选择电影语音的语言。

7.4.6 刻录软件 Brasero

Brasero 是一款多媒体软件，在顶部面板上选择“应用程序”|“多媒体”|“电影播放机”菜单命令，启动应用程序，显示 Brasero 的主界面如图 7-30 所示。

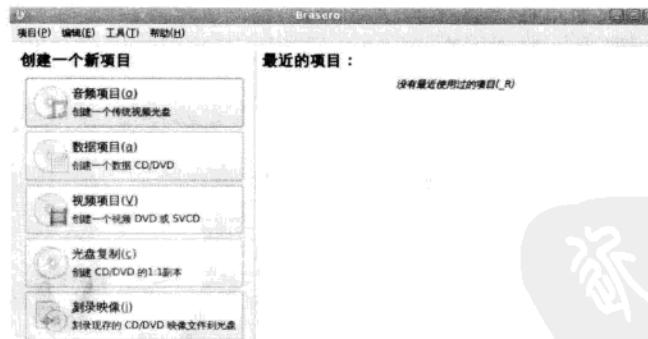


图 7-30 Brasero 主界面

它是一款 GTK+ 界面的光盘烧录软件，其简易、独特的功能使用户能够轻松而快速地在 Linux 下烧录光盘。在 GNOME 2.26 中，GNOME 对 Brasero 做了很大的修改，Brasero 添加了刻录 DVD 和 VCD 的功能，并且可以使这些功能添加到其他的应用程序中。用户可一边观看电影一边通过电影播放机来刻录视频。这是 GNOME 的 Brasero 能够做到的。Brasero 提供了光盘封面设计的功能，提供了当光盘刻录完毕之后自动弹出光盘功能。Ubuntu 系统安装的时候就默认安装这个软件，很适合家庭用户，方便快捷，使用简单。

7.4.7 邮件程序 Evolution

Evolution 支持常规的邮件功能，如收发邮件、邮件过滤、数据备份等，另外，Evolution 也支持用户插件功能，Evolution 默认内置了十几种用户插件，如 Bogofilter 垃圾邮件检测、附件提醒，以及新邮件提醒等。Evolution 作为免费的邮件收发软件，可以收发 163、yahoo、gmail 等邮箱的邮件。

Evolution 还集成了联系人、任务、备忘和日历功能，方便用户进行日常工作和生活，并且 Evolution 可以从 Windows 下的 Outlook 程序中导入用户的联系人和信件，而且 Evolution 与 Linux 下的其他程序之间也很容易迁移信件，如 Thunderbird 应用程序。

在顶部面板上选择“应用程序”|“办公”|“Evolution 邮件及日历”菜单命令，启动 Evolution 应用程序，打开主界面如图 7-31 所示。



图 7-31 Evolution 主界面

7.4.8 游戏

GNOME 中有许多小游戏程序。在“应用程序”|“游戏菜单”上用户可以找到很多小游戏，如图 7-32 所示，其中包括纸牌游戏（图 7-33）、麻将（图 7-34）和国际象棋（图 7-35）等。



图 7-32 game menu

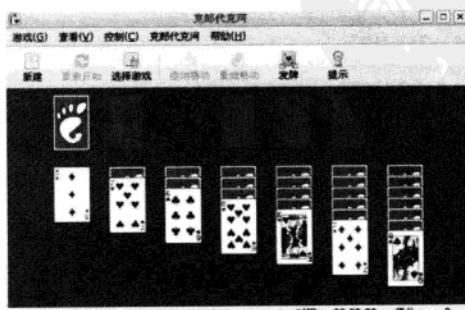


图 7-33 纸牌游戏

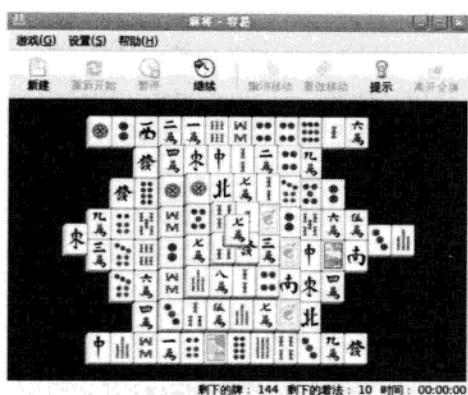


图 7-34 麻将



图 7-35 国际象棋

7.5 小结



GNOME 桌面环境是基于 GTK+开发包而开发的，提供给用户一个免费的，简单易用的操作桌面。GNOME 桌面为其下的基础工具提供图形的设置界面，并把系统的配置功能放在“系统”菜单下，简化了用户配置系统程序的操作，方便用户配置和管理系统。

随着 GNOME 桌面环境的发展，用户逐渐转向使用 Linux 操作系统。从 Windows 桌面系统转过来的 Linux 用户可以在 Ubuntu 中找到许多 Windows 下软件的替代软件，并且同样方便地进行日常工作、生活和娱乐活动。

GNOME 有着强大的开发团队，所有 GNOME 项目都可以在官方网上找到，并且吸引着很多 GNOME 的支持者。更多 GNOME 软件可查看网址：<http://projects.GNOME.org/>。

在下一章将讲述另外一个桌面环境 KDE，并简单说明 Kubuntu 发行版本，介绍 KDE 的组成，KDE 下的常用软件。

7.6 习题



1. 描述 GNOME 的历史。
2. GNOME 有什么优点？
3. 在顶部面板中添加粘贴标签小程序，并移动它的位置，移动到合适位置之后，锁定它的位置。
4. 在桌面上添加应用程序 pidgin 的启动器，其中 pidgin 应用程序的路径为 /usr/bin/pidgin。
5. 在“应用程序”下添加“mine”|“bash”菜单命令，其中 bash 为新建的启动器名称，启动器的类型为终端的应用程序，启动器的命令路径为 /bin/bash。

Linux

第 8 章

KDE 桌面环境

KDE 是 K 桌面环境（K Desktop Environment）的英文缩写，它是一种运行于 UNIX 以及 Linux、FreeBSD 等类似于 UNIX 操作系统上面的著名的自由桌面环境，整个系统采用的是 Qt Software 公司所开发的 Qt 开发包。该项目的目的是提供基本的桌面功能和日常必需的应用程序，以及提供开发者编写独立的应用程序的工具和文档。

本章将介绍 KDE 桌面环境，讲述 KDE 的历史，简述 Kubuntu 发行版本的安装和使用，讲述 KDE 桌面环境下常用的应用软件。





8.1 KDE 简介

KDE 是 K Desktop Environment 的英文缩写，是一套完全遵循开放源码宗旨的组件化开源桌面环境，是完全自由和免费的，整个系统采用的都是 Qt Software 公司所开发的 Qt 开发包。KDE 在设计之初就致力于为用户提供一种简便的方式来使用 Linux 与其他 UNIX 系统。随着 KDE 的发展，KDE 项目吸引了来自世界各国的软件开发人员参与其中，KDE 桌面也逐渐得到了一些 Linux 主要发行版本的支持。

8.1.1 KDE 起源

在 KDE 还没有出现之前，用户都使用 X11 桌面。传统的 X11 并不友好，对普通用户来说存在以下缺点：

- 没有一个基于桌面配置的简易的人机对话机制。
- 没有统一的应用程序帮助系统。
- 没有通用的应用程序开发框架。
- 没有一个复合文档框架。
- 缺少应用程序级的网络透明性。
- 编写 X11 程序非常困难枯燥。

用户是无法忍受这样的操作界面的。为了解决上述问题，德国人 Mathias Ettrich 在 1996 年 10 月创建 KDE 项目，确切时间是在 10 月 14 日公布的。当时 Mathias 就读于蒂宾根大学，他忧心于 UNIX 桌面没有一个应用程序外观、感受或工作方式和其他程序一样。他建议不仅是建立一套应用程序，更是一个桌面环境，用户可以得到相同的外观、感受和工作方式。他还希望这个桌面易于使用及更人性化。他在 Usenet 发表的文章引发了大量的反响，而 KDE 计划就此诞生了。KDE 的发布，赢得了用户的普遍赞誉，并且开始有大批的软件工程师致力于它的发展。

补充说明

X Window (有时称为 X11 或 X)，是一个对网络透明的客户 / 服务器架构的图形窗口系统，之后变成 UNIX、类 UNIX，以及 OpenVMS 等操作系统所一致适用的标准化软件工具包及显示架构的运作协议。

1984 年 MIT (麻省理工学院) 和当时的 DEC 公司合作开启了在 UNIX 系统上开发一个分散式的视窗环境—X Window System 项目。MIT 和 DEC 的目的只在于为 UNIX 系统设计一套简单的图形框架，使 UNIX 工作站的屏幕上可显示更多的命令，对于 GUI 的精美程度和易用程度并不讲究，毕竟那时候能够熟练操作 UNIX 的都是些习惯命令行的高手，根本不在乎 GUI 存在与否。1986 年，MIT 正式发行 XWindow，此后它便成为 UNIX 的标准视窗环境。

X11 是 1987 年发布的版本协议，由于它只是一个工具包及架构规范，它本身并无实际参与运作的实体，所以必须有人依据此标准进行开发撰写。目前，依据 X11 的规范架构所开发撰写成的最为普遍且最受欢迎的一个实体是 X.Org。X11 系统并不是一套可运行的、完整的桌面环境，它不能向用户提供一个外观美丽、操作简单的桌面环境。

8.1.2 KDE 的发展

KDE 项目创建于 1996 年 10 月，并公布于众。Mathias Ettrich 创建 KDE 项目之后，很多志愿者都参与到项目中，在 1997 年，他和志愿者们为 KDE 添加了一些应用程序。KDE 团队于 1998 年 7 月 12 日发布了 KDE 1.0。在发布公告中 KDE 团队如此概述了这个专案及创造原因：

“KDE 是一个用于 UNIX 工作站的网络透明的现代化桌面环境。KDE 不断探索以满足 UNIX 工作站需要一个易于使用的桌面，类似 MacOS 或 Windows95/NT 下的桌面环境。我们相信 UNIX 操作系统是当今可用的最好的操作系统。事实上多年来，UNIX 在信息技术专业已经成为无可争议的选择。当提到稳定性、可扩展性和开放性，没有什么可以和 UNIX 竞争。但是，由于在 UNIX 上缺乏易于使用的现代化桌面环境阻碍 UNIX 成为办公室和家庭中普通计算机用户的桌面系统。

现在 KDE 是 UNIX 上可用的且易于使用的现代桌面环境。结合一个自由的 UNIX 如 Linux，UNIX/KDE 组成一个完全自由和开放的计算机平台，完全免费提供给任何人，包括对源代码的任何修改。虽然它总是会有改进的空间，我们相信已经提供一个当今一些较常见的和商业操作系统/桌面组合的合适的替代选择。我们希望 UNIX/KDE 组合将最终实现开放、可靠、稳定和专利自由的计算机环境。”

在 KDE 的发展中，KDE 带给广大用户如下的体验：

- 它提供了一个美观的现代化桌面。
- 它提供了一个具有完整的网络透明性的桌面。
- 它提供了一个方便的集成帮助系统，对 KDE 桌面及其应用程序帮助的一致化访问途径。
- 所有的 KDE 应用程序都具有统一的视觉观感。
- 它拥有标准化的菜单、工具栏、键盘绑定、颜色样式等。
- 它支持国际化，已拥有 60 余种语言的翻译。
- 它集中化组织的对话框系统，由具体的桌面配置来运作。
- 它拥有大量优秀的 KDE 应用程序。

8.1.3 KDE 前进动力和保障

一个庞大的项目，需要有一定的支持才能稳健发展。KDE 项目并不是一个人的项目，或是一个特定组织的项目，它是全世界都在参与的项目。我们先看看这些数字：现在的 KDE SVN 代码仓库已经储存了超过 400 万行的代码；超过 800 名贡献者在协助进行 KDE 的开发；独立的翻译小组大约有 300 人；仅在 2002 年 5 月，据统计就有 11014 次 CVS 代码提交；KDE 在 12 个国家有 17 个以上的官方 WWW；KDE 在 39 个国家有 106 个以上的官方 FTP 镜像。

庞大的项目要有一个合理的管理方法。KDE 社区试图努力地维持一个健康的探讨环境，采用了例如召开只面向核心开发者的内部会议频道，并且限制对核心开发邮件列表 kde-core-devel 的写入权限，只有实干的开发者才能参与其中，打消那些偏执的看客们的干扰。当然，kde-core-devel 邮件列表的权限仍然对所有人开放。

KDE 项目是依靠互联网上诸多牺牲自己业余休息时间的软件工程师的贡献而发展进步



的。KDE 的开发者拥有独占一个子项目的权利，也没有一个如同企业的明显等级制度。开发者有着更大的思考空间，会创造更多的创新点。每个开发者都按照自己的意愿去开发自己独占的那一个功能块和软件，独立实现一种创意的过程，并享受此过程中的开发乐趣。开发者的功能模块，会经核心开发团队的讨论，讨论通过则添加到 KDE 中，成为 KDE 中的一部分。

KDE 核心小组（The KDE Core Team）的加入资格仅仅取决于贡献者的工作成绩。每一个人都有机会成为核心小组的成员，但是这种特别的申请必须由开发者对 KDE 项目长时间的杰出贡献来支撑。KDE 核心小组可以决定项目的整体发展方向和发布日程表。和以 Linux 内核为代表的其他自由软件项目相反，KDE 社区里边没有一个所谓“慈善的领导”来主导对各种重大问题的决策，事实上 KDE 核心小组是由 20 余名开发者组成，所有重大事务都要经过民主的投票表决来确定。

KDE Free Qt 基金会成立于 1998 年，它由 Trolltech 公司和 KDE 协会共同组建，其目的是促使 Qt 工具包可以为自由软件尤其是 KDE 桌面环境未来的开发提供有效的可持续性担保。

8.2 Kubuntu 的简介



Kubuntu 是基于 KDE 桌面环境的一个非常友好的操作系统。作为 Ubuntu 项目的一部分，保持可以预测的 6 个月的发布周期，Kubuntu 对于所有人而言是 GNU/Linux 的发行版。

Kubuntu 9.04 基于 Linux 2.6.28 和 KDE 4.2.2，若用户喜欢使用 C++ 作为开发平台，可以选择 Kubuntu。因为 Kubuntu 能够很好地支持 C++ 程序，并为软件开发者提供了集成开发环境。

Kubuntu 团队对其使用者的承诺：

“Kubuntu 充分运用 Ubuntu 的坚固基础，并再加上最新版本的 KDE。我们是 Ubuntu 社群的一部分，使用他们的基建及支持。我们的任务是成为最好的 KDE 分发版本。”

Kubuntu 永远免费，即使是企业版也无须额外费用，我们要把最好的工作成果免费提供给所有人，一视同仁。

Kubuntu 包含了出色的翻译和无障碍环境架构，唯有如此，自由软件社群才能让尽可能多的人使用 Kubuntu。

Kubuntu 完全遵守开放原始软件开发的原则；我们鼓励人们使用、改进并散布开放原始码套件。”

Kubuntu 同样提供了 Live CD 版，用户可以使用 Live CD 版本尝试 Kubuntu 的使用。Live CD 的启动步骤与 Ubuntu 的 Live CD 系统的启动步骤一样，把 Live CD 插入到计算机的光驱中，提示系统操作时，选择“只是使用操作系统而不安装”选项，Live CD 引导程序会启动 Live CD 系统。关闭 Live CD 系统，如同在 Ubuntu 的 Live CD 系统关闭的操作，关闭系统之后，退出光盘即可。

Kubuntu 中提供了 1000 多款应用软件，其下软件包多以 K 开头。主要的应用程序有：文件管理器和网页浏览器 Konqueror、KDE 控制中心 Kcontrol Center、KDE 集成开发环境 Kdevelop 和 KDE 控制台 Konsole。

以下罗列 Kubuntu 中比较常用的一部分应用程序：

- 图形图像软件： KSnapshot 图片截图软件。
- 网络软件： Krfb 桌面共享软件、 Kmail 邮件客户端程序、 Qussal IRC 软件、 Konqueror 网页浏览器。
- 多媒体软件： K3b 刻录软件； KMix 声音混合器软件； Dragon Player 视频播放软件； Amarok 音频播放软件。
- 办公软件： KaddressBook 地址簿软件； Kontact 个人信息管理软件； OpenOffice 办公软件套件。
- 系统软件： Kuser 用户管理软件； Konsole 模拟终端。
- 附件软件： Ark 压缩解压软件； Kate 高级文本编辑器； KNote 标签程序； SpeedCrunch 计算器。

8.3 在 Ubuntu 下安装 KDE 桌面



Kubuntu 可以单独安装到磁盘中，亦可作为 Ubuntu 的软件包安装到 Ubuntu 系统。Kubuntu 安装到磁盘中的方法，与 Ubuntu 安装到磁盘中的方法一样。现在讲述后一种情况下，怎么安装 KDE 桌面环境。Ubuntu 9.04 的发行版本包中包含了 KDE 桌面所需的程序包。

8.3.1 新立得安装 KDE 桌面

Ubuntu 9.04 默认的桌面环境是 GNOME 桌面系统，假设此时系统的用户名为 dranyu，并且已经登录了 Ubuntu 系统，使用新立得软件包管理器安装 KDE 桌面环境的具体安装步骤如下：

(1) 选择顶部面板的“系统” | “系统管理” | “新立得软件包管理器”菜单命令，打开新立得软件包管理器，在快速搜索栏中输入 Kubuntu，就可以看到搜索结果如图 8-1 所示。

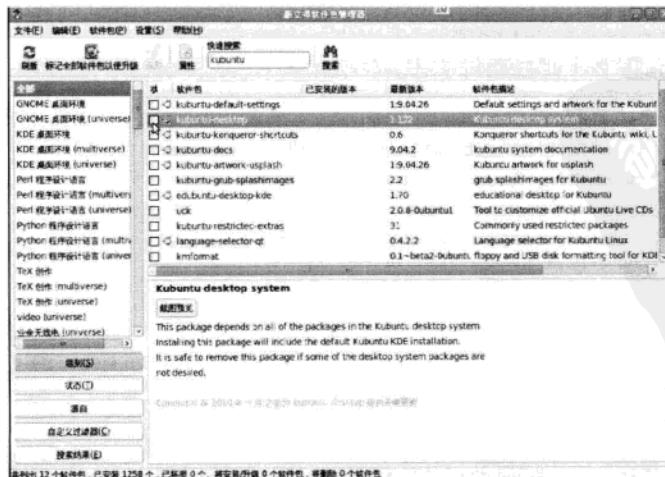


图 8-1 选择 KDE 桌面软件包



(2) 双击 Kubuntu-desktop 软件包，弹出如图 8-2 所示的提示窗口。新立得自动检测到 Kubuntu-desktop 软件包所依赖的软件包，并将 Kubuntu-desktop 软件包及其相关联的软件包在图 8-2 中罗列出来。单击“标记”按钮，关闭提示框，返回到新立得软件包管理器主界面，此时所需安装的软件包都以绿色标记。

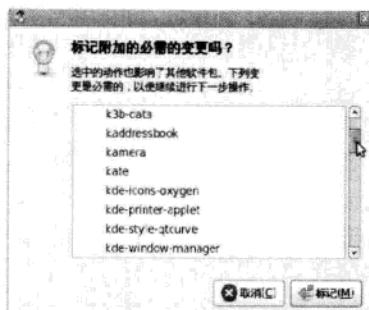


图 8-2 提示标记关联的软件包

(3) 在新立得软件包管理器主界面上，单击工具栏上的“应用”按钮，弹出摘要对话框如图 8-3 所示。此处罗列出即将被安装的软件包，单击“显示细节”按钮，将显示此次需要安装的所有软件包名称、安装所需下载软件包的大小、需占用磁盘空间的大小。

技术提示

若用户不期望新立得软件包管理器现在就安装 Kubuntu-desktop 软件包，可选择“仅下载软件包”选项，这样新立得软件包管理器就会把已作标记的软件包下载到 /var/cache/apt/archives 目录，以备以后安装 Kubuntu-desktop 软件包时使用。

(4) 单击“应用”按钮，弹出正在下载软件包文件对话框如图 8-4 所示。



图 8-3 安装摘要



图 8-4 下载软件包

(5) 软件包下载完毕之后，新立得自动关闭正在下载软件包文件对话框，并且弹出“Debcnf 运行于 dranyu-desktop”对话框如图 8-5 所示，设置默认的显示管理器。在下拉列表框中选择 kdm 选项，单击“前进”按钮，弹出正在应用变更对话框如图 8-6 所示，开始安装 Kubuntu-desktop 软件包，以及它所依赖的软件包，并把 kdm 作为默认的登录对话框桌面系统。



图 8-5 设置默认显示管理器

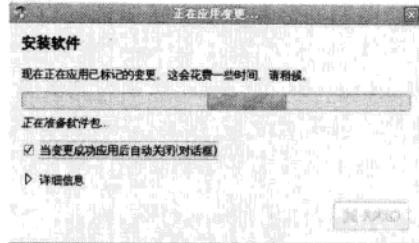


图 8-6 安装软件



图 8-7 变更已应用

(6) 新立得软件包管理器的操作从下载软件包转为安装软件包。软件包的安装需要一段时间，软件包安装完毕之后弹出变更已应用的提示框如图 8-7 所示。

通过查看详细信息可以了解软件包 Kubuntu-desktop 的安装情况，以及系统安装的操作过程。这一提示表示 KDE 桌面已经安装到了 Ubuntu 系统中。

(7) 单击“关闭”按钮关闭提示对话框，返回到新立得软件包管理器的主界面，此时已安装的 Kubuntu-desktop 软件包已被新立得使用绿色方格标记软件包的状态为已安装。

(8) 关闭新立得软件包管理器，重启计算机。由于选择 kdm 作为默认显示管理器，计算机重启后，KDE 桌面式的登录窗口界面如图 8-8 所示。

此时表示 KDE 桌面环境已成功安装到 Ubuntu 系统中。在会话类型的下拉列表框中选择 KDE 选项，然后在密码文本框中输入用户 dranyu 的密码就可以登录到 KDE 桌面，如图 8-9 所示。

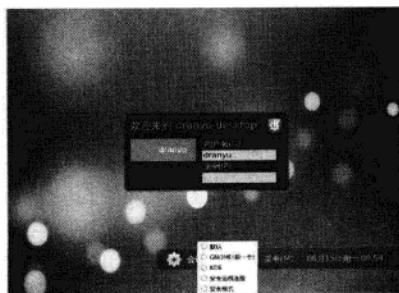


图 8-8 KDE 登录界面



图 8-9 KDE 桌面



补充说明

在第 5 个步骤中，我们需要设置默认的显示管理器。显示管理器用于管理系统登录会话的应用程序，提供登录屏幕的界面配置和自动登录配置等功能。gdm 和 kdm 是两个不同风格的图形化界面的显示管理器，控制着系统的登录屏幕。gdm 是 GNOME 桌面环境的一个重要组成部分，而 kdm 是 KDE 桌面环境的一个重要组成部分。Ubuntu 系统中默认桌面环境是 GNOME，因此 gdm 是默认的显示管理器。然而安装 KDE 桌面后，Ubuntu 系统就拥有两个显示管理器 gdm 和 kdm，需要选择其中之一作为默认的显示管理器。但无论 Ubuntu 使用哪种显示管理器程序登录系统，登录时都是使用 Ubuntu 系统已定义的用户名和密码。

8.3.2 命令行安装 KDE

除了使用新立得软件包管理器安装 KDE 桌面外，用户还可以使用终端命令来安装 KDE 桌面环境，具体操作步骤如下：

(1) 更新软件源，执行命令：

```
sudo apt-get update
```

(2) 使用 apt-get 命令安装 Kubuntu-desktop 软件包，在终端输入以下命令：

```
sudo apt-get install kubuntu-desktop
```

(3) 输入命令之后，按下回车键，开始执行命令。接下来的操作与使用新立得软件包管理器安装 Kubuntu-desktop 软件包的操作一致。

技术提示

sudo 命令表示获取 root 管理员的执行权限，使得当前登录用户拥有权限执行 apt-get 程序安装 Kubuntu-desktop 软件包。apt-get 命令是一个软件包管理器，在后续章节会详细介绍。apt-get 会自动检测与 Kubuntu-desktop 软件包有关联的软件包，并且下载和安装这些关联的软件包。

8.4 KDE 的面板和桌面



KDE 的面板（Panel）是 KDE 界面的一个重要组成部分。KDE 面板如同 GNOME 面板，使用方便、灵活，可以根据用户的需求放置应用程序和菜单。在 Kubuntu 中，KDE 的桌面和部件板是同一概念。KDE 桌面是一个可放大、缩小，可移动的桌面。

8.4.1 面板

KDE 默认在屏幕上放置了一个底部面板如图 8-10 所示。

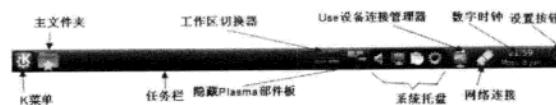


图 8-10 底部面板

● K 菜单

此 K 菜单是应用程序启动菜单，如同 Windows 下的启动菜单。单击 K 菜单图标弹出 KDE 桌面的主菜单如图 8-11 所示。用鼠标右键单击 K 菜单图标，弹出快捷菜单如图 8-12 所示，选择“切换为经典菜单风格”菜单命令，K 菜单切换为如图 8-13 所示的经典菜单风格。



图 8-11 K 菜单 Kickoff 风格



图 8-12 K 菜单快捷菜单

图 8-13 K 菜单经典风格

K 菜单中的“离开”选项卡包括了注销、锁定、切换用户、重新启动和关机等系统操作。

● 主文件夹



图 8-14 主文件夹窗口

主文件夹启动项能够让用户快速浏览主文件夹里边的内容，快速定位到主文件夹下的子文件夹。单击主文件夹图标按钮，弹出主文件夹浏览窗口如图 8-14 所示。

在浏览窗口中，用户可以浏览文件夹里边的文件，而且当用户单击左上角的“打开”按钮时，就可使用默认文件管理器 dolphin 中浏览地址栏上的文件夹下的文件。

● 任务栏

底部面板最长的空白处是任务栏。用户每启动一个应用程序，系统就会在任务栏面板上显示相应的应用程序最小化图标，表示应用程序已开始运行。在每个桌面工作区上运行的应用程序都会显示在此任务栏上，当用户单击应用程序的最小化图标时，应用程序被激活，而且应用程序所在的桌面工作区也被激活。

● 工作区切换器

工作区切换器面板工具，如同 GNOME 中的工作区切换器一样。切换工作区的方法有多种，其中包括：把鼠标放在工作区切换器上，滚动鼠标切换工作区；单击工作区切换器中的矩形切换工作区；按下“ $Ctrl+Alt+左右方向键$ ”组合键。

● 其他部件

系统托盘上放置已经启动并指定显示在系统托盘上的应用程序。在图 8-10 上显示目前系统托盘部件包括了语音控制、显示器、剪切板工具和更新提示。数字时钟显示系统的时间，单击数字时钟用户可以查看日历。设置按钮部件主要负责管理面板的属性。



8.4.2 操作面板和桌面

KDE 桌面环境使用 Plasma 程序管理着面板和桌面。

● 面板设置

底部面板最右边的图标是设置底部面板参数的设置按钮。单击此按钮，可以设置底部面板的长度、宽度和显示的方式等，如图 8-15 所示。



图 8-15 面板的设置

单击“屏幕边缘”按钮，将面板拖至屏幕的边缘，这样可改变面板的位置。

单击“高度”按钮，并拖动鼠标向上移动可以增加面板的高度，相反就会降低面板的高度。

单击设置菜单第二行上左右两侧的箭头，并用鼠标左右拖动箭头，就可以修改面板的长度。

● 添加部件

部件是可以运行在面板和桌面上的小应用程序。底部面板上的 K 菜单、主文件夹和任务栏等工具都属于部件。在 KDE 桌面环境中添加部件的步骤和方法如同在 GNOME 桌面上添加项目到面板的方法一样，单击图 8-15 上的添加部件按钮画出图标就可以向面板添加部件。

但在 KDE 环境桌面上除了能够在面板上添加项目外，它还允许用户向桌面添加部件，设置方法为：单击桌面右上角的设置桌面按钮，在显示的菜单上选择“添加部件”选项，之后的设置步骤与添加项目到面板的步骤相同。

● 添加壁纸

添加壁纸是为了设置美观的桌面背景。KDE 桌面环境提供给用户一个方便设置墙纸的图形管理界面，而且还集成了下载桌面壁纸的数据源服务器。将本地磁盘上的一张 5.jpg 图片设置桌面壁纸的具体步骤如下：



图 8-16 桌面设置

(1) 在桌面上，单击设置桌面按钮，在弹出的快捷菜单中选择“外观设置”菜单命令，弹出“桌面设置-Plasma 工作区”对话框，在壁纸选项框中，单击“类型”下拉列表，选择“图片”选项，然后单击图片设置栏后面的浏览图标，浏览文件夹获取指定壁纸图像文件“5.jpg”，然后单击“位置”下拉列表，选择“缩放和剪切”选项，操作结果如图 8-16 所示。对话框右边为壁纸的预览效果。

(2) 单击“应用”按钮，将改变应用到 KDE 桌面背景上，单击“确定”按钮，关闭“桌面设置”对话框，完成设置。

补充说明

KDE 桌面壁纸还可以从官方网站获取，单击“获得新壁纸”按钮就可以从 KDE 桌面环境的社区网站上下载新壁纸。网址为：<http://kde-look.org/>。

● 更换桌面主题

KDE 桌面主题包括颜色方案、面板的背景等。KDE 桌面环境允许用户定制个性化的桌面主题，具体的操作步骤如下：

- (1) 单击桌面右上角的设置按钮，在弹出的菜单中选择“外观设置”菜单，打开“桌面设置-Plasma 工作区”对话框，在桌面主题选项框中，单击主题后面的“新主题”按钮，打开“获得百宝箱！-Plasma 工作区”对话框，从主题列表中选中“Ateller”主题选项，单击“安装”按钮，系统自动安装桌面主题，当桌面主题安装完毕之后，原显示为“安装”按钮，变为“卸装”按钮，操作结果如图 8-17 所示。

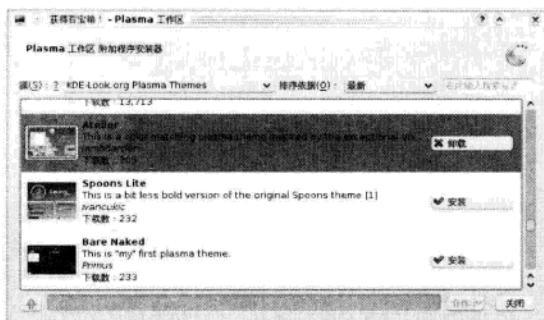


图 8-17 安装新主题

- (2) 单击“关闭”按钮，关闭“获得百宝箱！-Plasma 工作区”对话框，返回到“桌面设置-Plasma 工作区”对话框，然后单击“主题”下拉列表中的微调按钮，选择刚添加到名为 Ateller 的选项，如图 8-18 所示。

- (3) 单击“应用”按钮，执行修改桌面主题，主题应用于系统中，完成修改桌面主题的操作。



图 8-18 设置桌面主题

补充说明

在“获得百宝箱！-Plasma 工作区”对话框上，可以设置主题的来源。对话框中的主题列表框里列举了许多可安装的 KDE 主题。单击图 8-17 的主题列表框中项目后边的“卸装”按钮，可以卸装已安装在系统中的主题。



8.5 KDE 桌面环境的配置

KDE 桌面环境中为系统提供了一个集成所有系统设置的图形化配置程序，这个程序就是 Systemsettings（系统设置程序）。单击 K 菜单按钮，切换至“计算机”标签，选择“系统设置”菜单命令，弹出系统设置主界面如图 8-19 所示。



图 8-19 系统设置

在系统设置界面上有两个标签：常规和高级。在常规标签页上又分为感观、个人、“网络和链接”和计算机管理四个分类。

- 感观：此分类可以设置 KDE 桌面的外观、桌面、窗口行为和通知。
- 个人：此分类用于设置 KDE 桌面环境的地区和语言，以及制定默认应用程序。
- 网络和链接：此分类提供设置网络链接的功能，并且设置共享。
- 计算机管理：此分类设置 KDE 桌面的字体、显示器分辨率、键盘和鼠标的使用等。

在高级用户标签里用户可设置登录窗口、服务器管理器、会话管理器和硬件等配置。KDE 桌面环境的系统设置多数可以在此对话框中找到，下面来介绍一下 KDE 桌面环境常用配置。

8.5.1 修改时间日期

修改系统的时间和日期是使用操作系统时需要掌握的操作。接下来描述在 KDE 系统设置模块下修改时间和日期的具体方法。

- 在系统设置对话框中，选择“常规”选项，切换至“常规”选项卡，在此选项卡上双击“日期和时间”图标，打开“日期和时间-系统设置”对话框，在对话框中设置日期和时间，具体配置如图 8-20 所示，单击“应用”按钮就完成了修改系统日期和时间的任务。

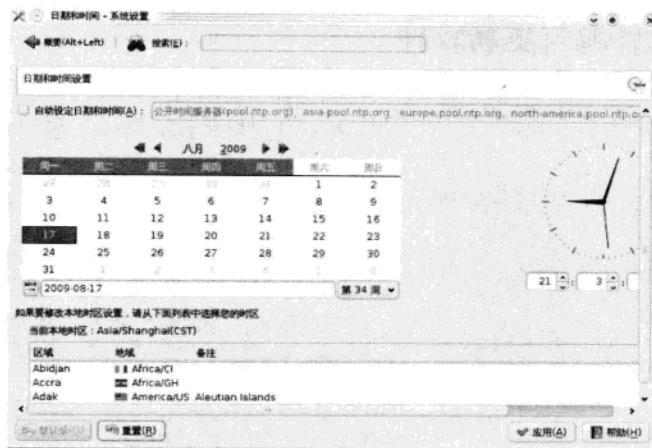


图 8-20 设置日期和时间

- 在图中用户还可选择时区，正确的时区有利于时间与世界时钟的同步。注意此处不能选择“自动设定日期和时间”选项，因为到目前为止，此选项所提供的时区信息，并不包括中国大陆。

8.5.2 修改显示器的分辨率

在第 6 章的时候，我们介绍了在 GNOME 桌面环境怎么设置显示器，在 KDE 桌面上设置显示器分辨率的步骤也差不多。这里就不详细简述设置显示器分辨率了，主要讲述如何启动显示器分辨率设置的程序，具体步骤如下：

在“系统设置”对话框，选择“常规”选项，切换至“常规”选项卡，双击“显示”按钮，进入设置分辨率对话框。在“大小”下拉列表框中选择“1024×768”选项，单击“应用”按钮即可。设置结果如图 8-21 所示。

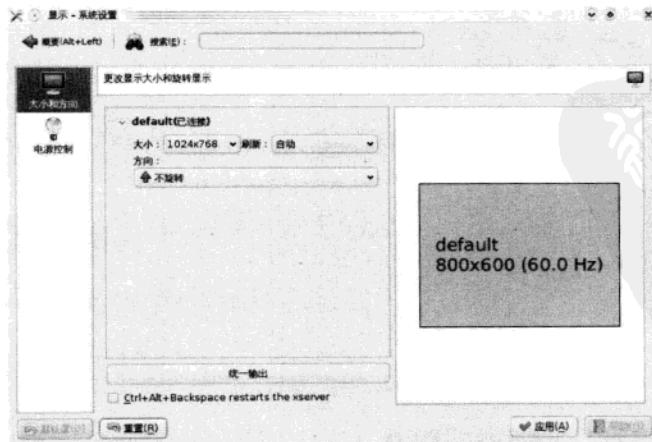


图 8-21 设置显示器分辨率



8.5.3 设置软件源与更新软件



图 8-22 设置软件源

如同 GNOME 桌面环境一样，KDE 桌面环境同样也需要设置软件源。Ubuntu 系列发行版本中，KDE 桌面环境设置软件源的方法和 GNOME 桌面环境的方法一样，具体设置方法为：在系统设置对话框中，选择“常规”选项，切换至“常规”选项卡，双击“添加和删除软件”按钮，进入到“添加和删除-系统设置”对话框，选择“设置”菜单命令，然后再单击右边的“Edit Software Sources”按钮，如图 8-22 所示，打开软件源对话框，之后的设置步骤和参数与第 6 章中的 6.8 节相同。

补充说明

在“添加和删除软件—系统设置”对话框中，选择“软件管理”菜单命令，可查看软件源中哪些软件包是可用的，可添加或删除指定的软件包。而选择“软件更新”菜单命令，可查看目前系统软件包可用更新的情况，并对更新提示做出反应，以更新系统。

8.5.4 设置默认的应用程序

使用 KDE 系统设置功能设置 Firefox Web Browser 为默认网页浏览器，具体操作步骤如下：

(1) 在系统设置对话框中，切换至“常规”选项卡，双击“默认应用程序”按钮，打开“默认应用程序—系统设置”对话框，在左边的菜单列表框中选择“Web Browser”选项，然后选中“在以下浏览器中”选项，如图 8-23 所示。

(2) 单击文本框后边的浏览项目按钮 [...]，打开选择应用程序的列表框，从应用程序列表框中选择“Firefox Web Browser”菜单命令，然后单击“确定”按钮，如图 8-24 所示。

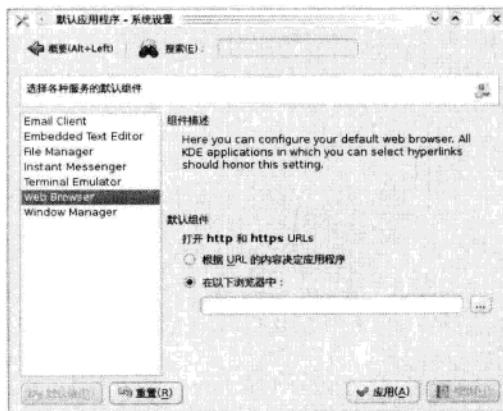


图 8-23 设置默认网页浏览器

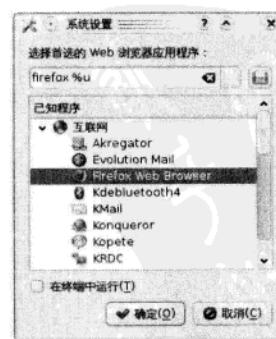


图 8-24 获取网页浏览器程序

(3) 然后单击“默认应用程序—系统设置”对话框中的“应用”按钮，完成设置操作。



8.6 KDE 桌面环境中常用软件

随着 KDE 项目的快速发展，面向 KDE 开发的软件越来越多。在 KDE 中国网站上列出了 KDE 的家族软件，网址为 <http://www.kdecn.org/kdesoft/kde3/>。在这里介绍几个在 Kubuntu 中默认安装的常用应用软件。

8.6.1 Konqueror 网页浏览器

Kubuntu 9.04 中使用 Konqueror 作为默认的网页浏览器。打开 K 菜单，选择“应用程序”|“互联网”|“网页浏览器”菜单命令启动 Konqueror，Konqueror 的主界面如图 8-25 所示。



图 8-25 Konqueror 网页浏览器

Konqueror 功能强大，启动和运行速度却很快，它在界面上看起来与 Firefox、IE 等其他网页浏览器相似，可以浏览文件、目录和网页内容。它遵循 HTML，支持 JavaScript、Java applets、CSS、SSL 等相关标准。

Konqueror 同时也是一个文件管理器和文件浏览器，它拥有标准文件管理器的功能，但它并没有成为 Kubuntu 9.04 的默认文件管理器。

8.6.2 Dolphin 文件管理器

Dolphin 文件管理是 KDE 4 的新文件管理器，并成为 KDE 4 的默认文件管理器。打开 K 菜单，选择“应用程序”|“系统”|“文件管理器”菜单命令启动 Dolphin 应用程序，默认主界面如图 8-26 所示。

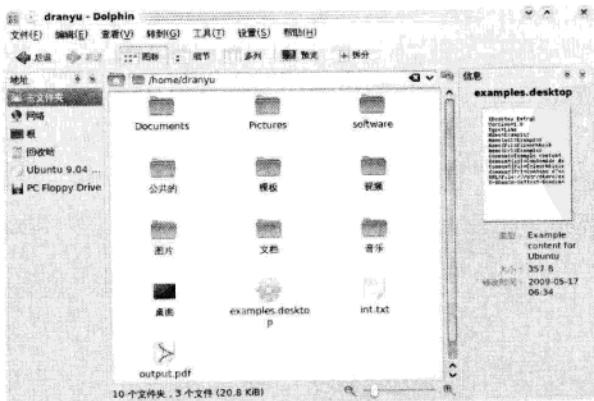


图 8-26 Dolphin 文件管理器

Dolphin 的设计专注于文件管理的易用性，它不像 Konqueror 既是网页浏览器又是文件管



理器，同时它与 Konqueror 共享了文件管理视图的功能代码，具备标准文件管理器的功能，如文件的浏览、定位、打开、复制及移动等操作。

8.6.3 K3b 烧录软件

K3b 是 Kubuntu 中的默认刻录软件，它提供 DVD、CD 和数据光盘等刻录功能。单击 K 菜单，选择“应用程序”|“多媒体”|“CD&DVD 刻录软件”菜单命令，启动 K3b 程序，K3b 主界面如图 8-27 所示。

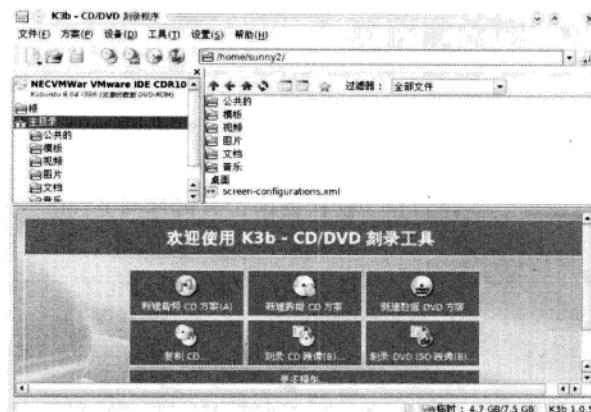


图 8-27 K3b 主界面

8.6.4 Kate 高级文本编辑器

Kate 高级文本编辑器是 KDE 桌面上默认的文本编辑器，它集成了文件系统浏览器，文本编辑器和终端，提供了方便的文件编辑管理，支持代码高亮突出显示和语法检查，支持多种模式编辑方式，如可切换至 vi 模式和覆盖模式编辑文本。

打开 K 菜单，选择“应用程序”|“工具”|“高级文本编辑器 Kate”菜单命令，启动 Kate，Kate 的主界面如图 8-28 所示。



图 8-28 Kate 高级文本编辑器

8.6.5 KMail 电子邮件客户端

KMail 是 KDE 中的电子邮件客户端。多数用户也许只会用到它的少部分功能，即设定一对 POP、SMTP 服务器完成常规的邮件收发工作。KMail 程序还具备以下的特点：

- (1) 在 KMail 中，用户可以设定多种发信身份和多个邮箱账户，并将不同身份和不同邮箱账户、密钥编号、信头签名、邮件夹等一一对应。
- (2) KMail 支持 IMAP 协议。IMAP (Internet Message Access Protocol) 是一种新式的邮件访问协议，可以使得用户上传邮件如同移动本地磁盘的文件一样简单。
- (3) KMail 接受 OpenPGP 与 S/MIME 两套加密算法，内置一套全面的加密策略配置机制，在编写信件时只靠一键即可完成签名和加密步骤。
- (4) KMail 可针对邮件规范内的各种属性字段实行内容匹配，并分门别类设定过滤后的操作。
- (5) KMail 可以检测系统中已安装的第三方反垃圾邮件/反病毒程序，并自动和它们协调工作，筛选恶意信件。

单击 K 菜单，选择“应用程序”|“互联网”|“邮件客户程序 KMail”菜单命令，启动 KMail 应用程序，如图 8-29 所示。



图 8-29 KMail 应用程序

8.6.6 Konsole 模拟终端



图 8-30 Konsole 模拟终端

Konsole 是 KDE 桌面环境模拟终端，支持多标签、国际化编码、双向文本渲染、ZModem 文件上传等功能，同时提供了多种样式风格，而且还可以作为插件被嵌入到其他应用程序中。单击 K 菜单，选择“应用程序”|“系统”|“终端”菜单命令，启动终端，其操作界面如图 8-30 所示。

8.6.7 KPackageKit 软件包管理器

KPackageKit 软件包管理器是 Kubuntu 下默认软件包管理图形界面。单击 K 菜单，选择



“应用程序” | “系统” | “软件管理”菜单命令，弹出 KPackageKit 软件包管理器操作界面如图 8-31 所示。

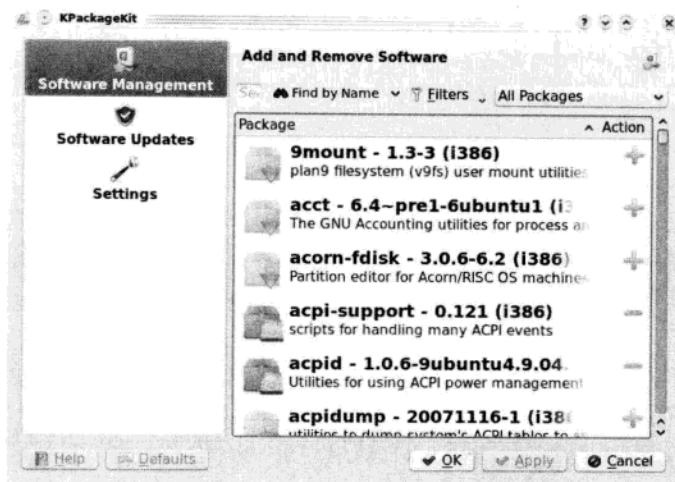


图 8-31 KPackageKit 软件包管理器

8.6.8 System Monitor

System Monitor（系统监视器）如同 Windows 下的任务管理器。单击 K 菜单，选择“应用程序” | “系统” | “系统监视器”菜单命令，打开系统监视器主界面如图 8-32 所示。

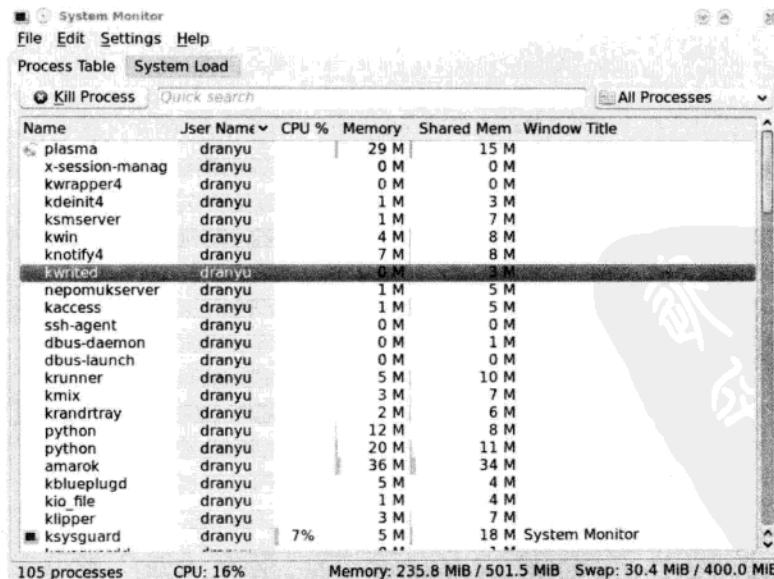


图 8-32 System Monitor



8.7 小结

本章主要介绍了作为 Kubuntu 9.04 默认桌面的 KDE 桌面环境。KDE 桌面环境是基于 Qt 开发包的，克服了传统的 X11 桌面的缺点，致力于为 Linux 用户营造一个操作简单、界面友好的桌面环境。

Kubuntu 9.04 使用了 KDE 4.2 版本桌面环境，大大提高了 Kubuntu 的性能，给用户提供了更多的视觉体验，从文件管理到系统管理，都给用户提供了一个简洁的操作界面，功能设置简单，使用方便。KDE 桌面环境上的应用软件越来越多，完全可以满足用户的日常学习、工作和娱乐所需。

在下一章中，我们将继续讲述桌面系统，主要讲述另一个比较主要的桌面环境 Xfce 桌面，它是 Ubuntu 另一个衍生版本 XuNbuntu 采用的默认桌面系统。

8.8 习题



1. KDE 项目是由谁发起的？KDE 是基于何种语言开发的？使用的是何种开发库？
2. KDE 默认程序分为哪几大类？都包含了哪些常用应用软件？
3. 使用新立得软件包管理器添加 kubuntu-desktop 软件包，安装完毕之后，重启 Ubuntu，使用 KDE 会话进入到 KDE 桌面环境中。
4. 设置 KDE 桌面的背景图片，为 KDE 桌面安装新桌面主题。
5. 向 KDE 桌面环境的底部面板上添加小应用程序。
6. 设置 KDE 桌面环境上显示器的分辨率。
7. 打开 Konsole 模拟终端，在终端的命令提示符输入并执行 ls 命令。
8. 先打开 Kate 高级文本编辑器，然后打开 System Monitor（系统监视器）查看在进程列表中是否存在 Kate 进程？

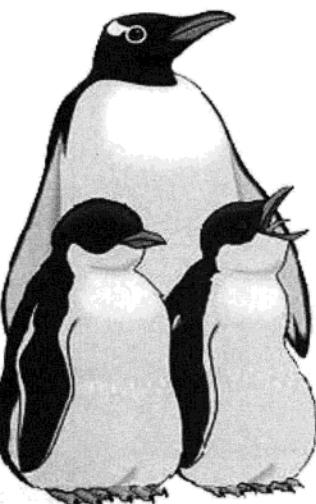
Linux

第9章

Xfce 桌面环境

Xfce 桌面系统是一个轻量级桌面环境，它提供了一个标准桌面系统的全部功能。它的宗旨是提高用户的工作效率，在节省系统资源的同时，能够快速加载和执行应用程序。Xfce 桌面适合硬件配置比较低的计算机。Xubuntu 9.04 采用了 Xfce 桌面系统环境作为默认的桌面环境系统。

本章主要介绍 Xfce 桌面环境的起源及其发展历程，继而介绍使用 Xfce 作为默认桌面系统的 Xubuntu，讲述在 Ubuntu 中如何安装 Xfce 桌面环境，简述 Xfce 桌面环境的核心组件与 Xubuntu 中常用应用软件。





9.1 Xfce 简介

Xfce 是一个用于类 UNIX 操作系统的轻量级桌面环境，它基于 GTK+ 2 开发包而开发，它的设计目标是快速小巧，对系统资源要求少，同时仍能做到外观华丽且易于使用。

Xfce 4 体现了传统 UNIX 系统中的模块化和可复用性，所有 Xfce 4 核心组件都可以进行独立构建和安装。

坚守标准是 Xfce 4 的一个主要设计目标，特别是在 Freedesktop.org 上定义的那些标准。这使得 Xfce 4 可以和为其他桌面环境编写的程序进行无缝的互操作，只要这些桌面环境也遵循相应的规范和标准。

9.1.1 Xfce 简史

网络开发设计者和嵌入式系统的设计师——Olivier Fourdan 于 1996 年开发了 Xfce 项目。他是以“XForms Common Environment”的前面第一个字母的缩写为 Xfce 命名的。在其发展过程中，经过了两次重写，之后的版本就不再使用原来的工具包了，可是 Xfce 这一名称保留了下来。Xfce 的开发时间比较长，这样也就保证了可以提供给用户一个可靠稳定的桌面系统。Xfce 到目前为止，主要有四个版本，目前 Xubuntu 使用了 Xfce4.4，在性能上得到了很大的提升。

第一个版本，Xfce 开始的时候使用 Xform 开发的只是一个简单的桌面运行环境，Fourdan 把它作为一个小的工具包发布到 ibiblio。那时 Xfce 只是一个工具包，并没有多大的处理功能。

第二个版本，Fourdan 为 Xfce 添加了第一个版本的窗口管理器 Xfwm，于 1998 年发布了 Xfce。Fourdan 要求把 Xfce 添加到 Red Hat Linux 中，可是由于 Xfce 是基于 Xform 的，而遭到 Red Hat 的拒绝。其原因是当时 Red Hat 只接受免费的开源应用程序，但是 Xform 是免费的，并不是开源的，仅供个人使用。同样的原因，Xfce 到了第三版本才被添加到 Debian Linux 系统中。

第三个版本 Xfce 3 于 1998 年发布。鉴于之前没有得到大公司的应用，而不能成为流行 Linux 版本的一部分，Xfce 的创建者 Fourdan 就对 Xfce 进行了一次全新改版，使用越来越流行的开发包 GTK+重新开发 Xfce。Xfce 3 使用 GNU GPL 发布，并且得到了 GTK 的支持和源语言支持，并且改良了可配置性能。于 2001 年 2 月，Xfce 被上传到了 sourceForge.net，那时版本为 Xfce3.8.1，真正成了一个免费开源的桌面系统。

第四个版本 Xfce4 于 2003 年发布。Xfce 4 使用了 GTK+2 库，性能得到了很大的提升。这时 Xfce 4 桌面系统被其他流行的 Linux 发行版本作为候选的桌面环境加入到发行包中。目前，Xfce 4 最新版本为 Xfce 4.6，于 2009 年 2 月 27 日发布，并作为 Xubuntu 9.04 默认桌面环境。

9.1.2 Xfce 的精神

Xfce 与其他如 GNOME、KDE 桌面不同，Xfce 主要目标是快速、轻量级，当然也追求易用。Xfce 4.6 体现了传统 UNIX 系统中的模块化和可复用性。



Xfce 是一个完全的自由软件。Xfce 的每个独立组件都是按 BSD 许可证、GNU 通用公共许可证或 GNU 库通用公共许可证之一进行发布。开发者可从网站 (<http://www.xfce.org/>) 上下载源代码，而且 Xfce 开发社区允许开发者访问 Xfce 版本控制的仓库，访问的 svn 地址为 <http://svn.xfce.org/svn/xfce>。

9.2 Xubuntu 的简介



Xubuntu（发音为 ZOO-bun-too）是一个 Ubuntu Linux 的官方衍生版，它基于 Xfce 桌面环境。Xubuntu 主要面向旧式电脑的用户和寻求更快捷的桌面环境的用户，它主要运行基于 GTK+库的程序。

Xubuntu 的开发团队如此定义 Xubuntu 的使命：

“The goal of Xubuntu is to produce an easy to use distribution, based on Ubuntu, using Xfce as the graphical desktop, with a focus on integration, usability and performance, with a particular focus on low memory footprint. The integration in Xubuntu is at a configuration level, a toolkit level, and matching the underlying technology beneath the desktop in Ubuntu. Xubuntu will be built and developed as part of the wider Ubuntu community, based around the ideals and values of Ubuntu.”

大概的意思是说：Xubuntu 的目标是开发一个基于 Ubuntu 的易用的衍生版本。重点在于集成性、可用性和执行性，更主要的是低的内存消耗。Xubuntu 的集成性能，体现在配置层面、工具包层面和匹配 Ubuntu 桌面下的底层技术的集成。Xubuntu 将作为 Ubuntu 社区开发的一部分，并基于 Ubuntu 的思想和价值。

Xubuntu 的第一个版本随 Ubuntu 5.04 而发布。从那时开始，Xubuntu 就使用了 Xfce4 版本的桌面，并与 Ubuntu、Kubuntu 和 Edubuntu 作为 Ubuntu 系列中的一员。从此之后，Xubuntu 也随着 Ubuntu 新版本的发布而发布，使用与 Ubuntu 相同的代码称号，同时采用当时最新发布的 Xfce 桌面。如目前 Xubuntu 8.10 版本长期支持的 Xfce 使用的 Xfce 4.4.2 桌面系统，而 Xubuntu 9.04 则使用了 Xfce 4.6 的版本。其他 Xubuntu 版本的发布时间版本情况，如表 9-1 所示。

表 9-1 Xubuntu 发布版本

发布代号	版本	发布时间	支持时间
Hoary Hedgehog	Version 5.04	2005 年 4 月	2006 年 10 月
Breezy Badger	Version 5.10	2005 年 10 月	2007 年 4 月
Dapper Drake	Version 6.06	2006 年 6 月	2009 年 6 月
Edgy Eft	Version 6.10	2006 年 10 月	2008 年 4 月
Feisty Fawn	Version 7.04	2007 年 4 月	2008 年 10 月
Gutsy Gibbon	Version 7.10	2007 年 10 月	2009 年 4 月
Hardy Heron	Version 8.04	2008 年 4 月	2011 年 4 月
Intrepid Ibex	Version 8.10	2008 年 10 月	2010 年 4 月
Jaunty Jackalope	Version 9.04	2009 年 4 月	2010 年 10 月

我们可以看到 Xubuntu 8.04 版本可获得长达 3 年的技术支持。

Xubuntu 9.04 发布于 2009 年 4 月 23 日，代号为 Jaunty Jackalope，与 Ubuntu 9.04 使用同样的代号。Xubuntu 9.04 使用的最新发布的 Xfce 桌面环境 Xfce 4.6 Xubuntu 9.04 得益于 Ubuntu 核心开发团队开发的改善启动时间的代码，使得 Xubuntu 9.04 的启动性能得到很大提高，达到了前所未有的快速启动时间。

9.3 在 Ubuntu 下安装 Xfce



在 Ubuntu9.04 下安装 Xfce4 是非常简单的事情。安装 Xfce 桌面系统，就如同在 Ubuntu 下安装其他软件包一样简单，具体操作步骤如下：

(1) 选择顶部面板的“系统”|“系统管理”|“新立得软件包管理器”菜单命令，打开新立得软件包管理器，在快速搜索栏中输入 Xubuntu 关键词，搜索结果如图 9-1 所示。

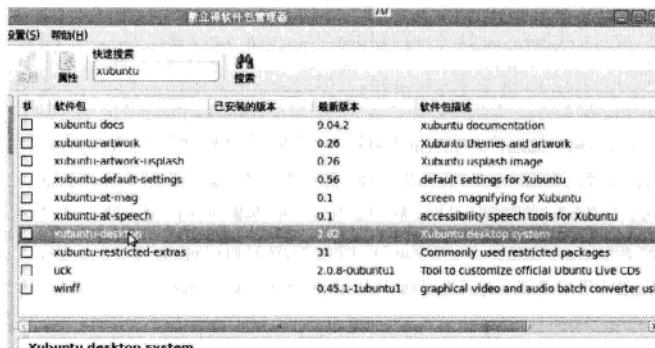


图 9-1 搜索 Xubuntu

(2) 选中 xubuntu-desktop 软件包记录，双击软件包，弹出提示对话框如图 9-2 所示。

(3) 按照默认的设置，单击“标记”按钮，返回到新立得主界面如图 9-3 所示，此时，新立得软件包管理器使用绿色标记需要安装的软件包。

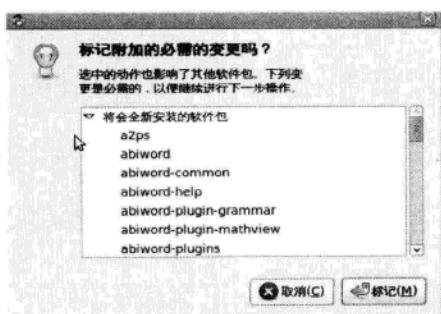


图 9-2 标记变更

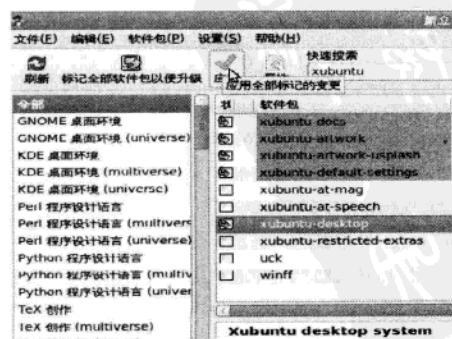


图 9-3 应用软件包的安装变更



(4) 确认无误之后，单击工具栏上的“应用”按钮，弹出提示对话框如图 9-4 所示。

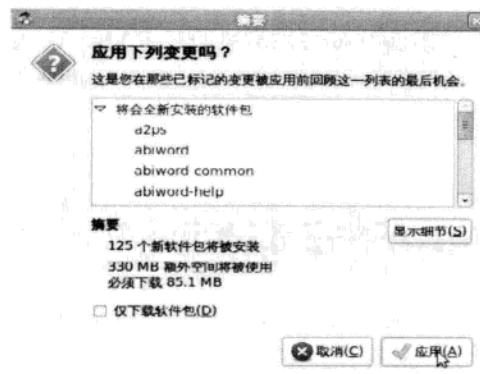


图 9-4 变更摘要

(5) 此对话框中罗列出安装 xubuntu-desktop 软件包时所需要安装的所有相关软件包。若选择“仅下载软件包”选项表示仅下载所有的软件包，而不安装软件包。此处取消选择此选项。

(6) 单击“应用”按钮，剩下的工作就交给了新立得软件包管理器处理。新立得软件包管理器开始自动下载软件包。当所需的软件包下载完毕之后，新立得执行安装软件包。当软件包安装完毕之后，新立得弹出“变更已应用”对话框如图 9-5 所示。

(7) 到此安装 xubuntu-desktop 软件包操作完毕，关闭新立得软件包管理器。

关掉新立得软件包管理器之后，重启计算机。在登录界面上单击“会话”下拉按钮，如图 9-6 所示，从列表框中选择“Xfce 会话”选项。在“欢迎”的选项框中，在“用户名”文本框输入用户名 dranyu，然后单击“确定”按钮，出现密码输入选项框，输入用户 dranyu 的密码，单击“确定”按钮，就可以启动 Xfce 会话。进入 Xfce 桌面环境的主界面，如图 9-7 所示。

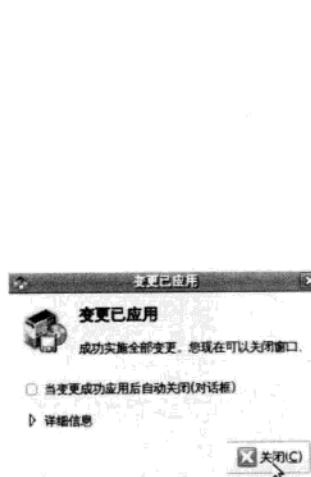


图 9-5 变更已应用

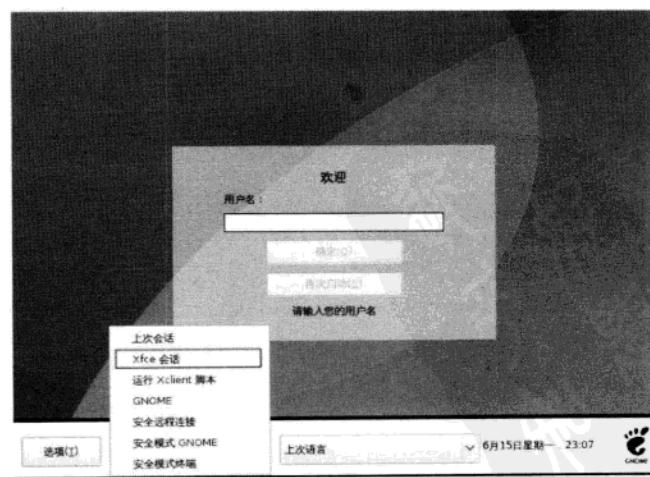


图 9-6 Xfce 登录界面

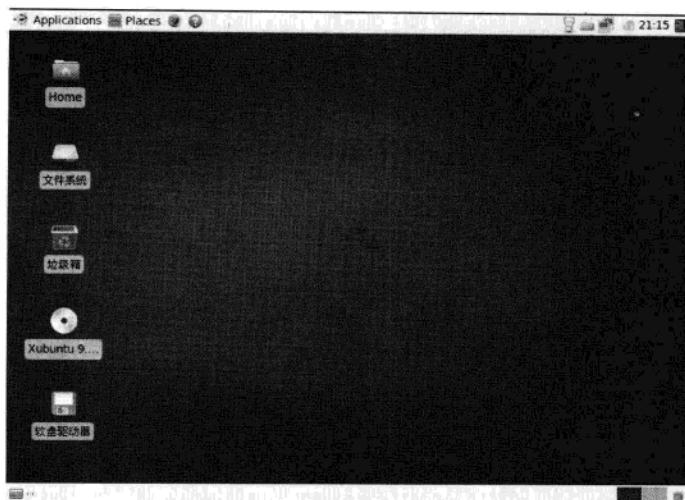


图 9-7 Xfce 桌面

除了使用新立得软件包管理器安装 Xubuntu，用户还可以通过在终端输入命令的方式安装。此步骤与安装 KDE 桌面环境的命令行步骤一样，只是命令不一样。在终端依次输入并执行如下命令：

```
sudo apt-get update  
sudo apt-get install xubuntu-desktop
```

执行命令后，系统自动寻找相关依赖的软件包，下载软件包并安装，安装过程与使用图形界面安装一样。

9.4 Xfce 面板和桌面



Xubuntu 中的面板和桌面与 Ubuntu 中的面板和桌面是一样的概念。认识 Xfce 的面板和桌面能够帮助用户快速掌握 Xubuntu 的使用技巧。

9.4.1 面板

Xfce 4 面板是 Xfce 4 桌面环境的一部分，用于放置程序启动器、菜单、时钟、桌面切換器以及其他。在 Xubuntu 中，默认有顶部面板和底部面板。

1. 顶部面板

顶部面板如图 9-8 所示，用户可观察到这个顶部面板与 GNOME 桌面很相似。



图 9-8 顶部面板



● Application (应用程序) 菜单

Application 菜单包含了系统设置菜单, 应用程序启动的菜单, 以及其他菜单。单击“Application”按钮, 弹出 Application 菜单如图 9-9 所示。选择此菜单中的“添加/删除”菜单命令可打开安装和删除第三方提供的软件包的软件包管理器。选择“设置”菜单命令弹出设置子菜单如图 9-10 所示, 此菜单列表显示了系统设置的所有菜单命令, 用户通过此菜单列表可设置系统外观、鼠标和键盘的使用、SCIM 输入法设置等。

● Places (位置) 菜单

Places 菜单放置了用户主目录、文件系统、软磁盘等磁盘目录的链接。

单击“Places”按钮, 弹出 Places 菜单如图 9-11 所示。两分隔符号中间的菜单项为用户主文件夹中的子文件夹名称。

其中“Search for Files”菜单命令调用一个名为 catfish 的查找文件的应用程序, catfish 应用程序可以在不同的文件目录下, 使用不同的查询命令如 find、locate 等来查找字符串或文件。

单击“最近的文档”菜单后面的下级菜单按钮, 可以看到用户近期使用的文件或浏览过的文件夹。

● 其他小程序

单击顶部面板的帮助文件快速链接的图标按钮, 可以打开 Xfce 桌面环境的帮助文档。从帮助文档中, 用户可解决在使用 Xfce 过程中遇到的部分问题。



图 9-9 Application 菜单



图 9-10 系统设置子菜单

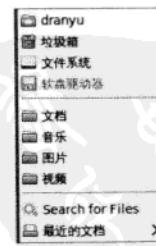


图 9-11 Places 菜单

单击动作按钮图标, 可执行系统退出、注销、切换用户、重启和关闭系统功能。

2. 底部面板

Xfce 桌面环境的另一个默认的面板为底部面板, 如图 9-12 所示。

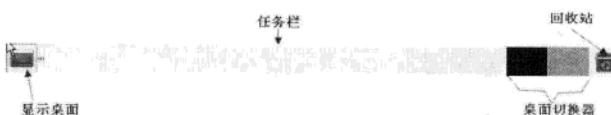


图 9-12 底部面板

底部面板与 GNOME 中的底部面板一样，最左边为显示桌面，最右边有一个工作切换器和一个垃圾箱。空白处为任务栏，用于显示已打开的应用程序的最小化图标。把鼠标放置在任务栏上，滚动鼠标的中键就可以激活任务栏上的应用程序。而将鼠标放置在桌面切换器上，滚动鼠标的中键就可以切换工作区桌面。

9.4.2 桌面

在图 9-7 中，除去顶部面板和底部面板后的区域我们称之为桌面。桌面的位置指向用户主目录下的桌面文件夹，即桌面上的文件或是启动器都保存在用户主目录下的桌面文件夹下（~/桌面）。桌面上放置了“文件系统”、“Home”和“垃圾箱”等图标。桌面作为 Xfce 的一个组成部分，其的管理程序是 Xfdesktop 组件（桌面管理器）。使用桌面管理器可以设置桌面的快捷菜单和中键菜单，还可设置桌面的背景。

9.4.3 操作面板和桌面

Xfce 桌面环境如同 GNOME 桌面环境，同样提供了个性化设置面板和桌面的功能。

- 添加新项目

Xfce 桌面环境提供了各种各样的小程序给用户，这些小程序可以运行在面板上。用鼠标右键单击面板上的任意位置，选择快捷菜单中的“添加新项目”菜单命令，就可以打开添加项目窗口，其添加方法如同 GNOME 桌面环境里的添加方法。

- 自定义面板

Xfce 的面板操作都集成在一个面板图形化管理工具中。底部面板的任务栏，弹出快捷菜单如图 9-13 所示，选择“自定义面板”菜单命令，弹出面板管理窗口如图 9-14 所示。此窗口提供了添加面板、删除面板、调整面板的功能。

添加面板的方法为：单击面板管理窗口中的添加图标按钮，就可以在桌面上添加一个拥有默认启动器的面板。



图 9-13 快捷菜单



图 9-14 面板



删除面板的方法为：在面板下拉列表中选择需要删除的面板名称，然后单击后面的删除图标按钮，相应的面板就会被删除。

调整面板参数的方法为：在面板下拉列表中选择需要操作的面板名称，然后在外观和位置选项框中设置相关参数，相应的参数就会立刻作用于选定的操作面板。

与 GNOME 有所不同的是，Xfce 在面板中弹出的菜单上并没有提供直接删除和添加面板的菜单项。

- 设置桌面背景图片

一张赏心悦目的图片可改变用户的视觉感受。设置桌面背景的具体操作步骤如下：

(1) 用鼠标右键单击桌面，弹出快捷菜单如图 9-15 所示，选择“桌面设置”菜单命令，弹出桌面对话框，切换至“背景”选项卡，然后单击添加图标按钮，从本地磁盘中选择自己喜欢的图片如“5.jpeg”，添加到桌面对话框的图像列表中，如图 9-16 所示。



图 9-15 快捷菜单



图 9-16 “桌面设置”对话框

(2) 在桌面对话框中，选中图像列表中的 5.jpeg 选项，桌面图像就会立刻改为 5.jpeg 图像，在风格下拉列表中选择“自动”选项，选中图像选项框中的“单个图片”选项。对于“亮度”和“饱和度”滑动条，用户可自行调节，桌面背景的效果会随着操作而不断改变。在图像列表中，Xfce 提供了多张默认的桌面背景图片。

(3) 设置了必要的参数后，单击“关闭”按钮，完成添加桌面背景图片的操作。

- 设置中键菜单

目前的鼠标多数包括三个键：左键、中键和右键。Xfce 桌面除了可以使用左键和右键外，还可以使用中键显示桌面的菜单。设置中键菜单的具体操作步骤如下：

(1) 用鼠标右键单击桌面，在弹出的快捷菜单中选择“桌面设置”菜单命令，弹出桌面对话框，选择“菜单”选项卡，然后在窗口列表菜单选项框中选中“按中键显示窗口列表菜单”选项，设置相关参数，如图 9-17 所示。

(2) 单击“关闭”按钮，完成设置中键菜单操作。此时在桌面上按下鼠标中键，弹出窗口列表菜单如图 9-18 所示。

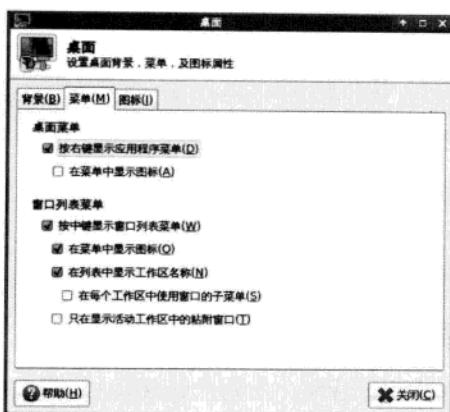


图 9-17 设置中键菜单



图 9-18 中键菜单

技术提示

使用中键菜单能够快速切换工作桌面，切换已打开的应用程序。在图 9-18 中，工作区 1 和工作区 2 表示目前桌面环境上有两个工作区（或叫桌面），单击工作区的名称就可以切换工作区。工作区下面的菜单项，表示当前运行在工作区中的应用程序，单击这些应用程序的名称就可切换至应用程序操作窗口。选择“添加工作区（A）”菜单命令，为系统添加一个工作区；而选择“移除工作区（R）2”菜单命令表示删除工作区 2。

● 设置桌面上的图标

若用户希望自己的桌面简洁，可以设置不显示默认的图标，具体设置步骤如下：

(1) 用鼠标右键单击桌面，在快捷菜单中选择“桌面设置”菜单命令，然后选择“图标”选项卡，在外观选项框中，从图标类型下拉列表中选择“文件/启动器图标”选项，取消选择默认图标选项框中的选项，如图 9-19 所示。

(2) 单击“关闭”按钮，完成去掉默认图标的操作。



图 9-19 取消选择默认图标



9.5 Xfce 的系统设置

Xfce 4 设置管理器提供了对所有 Xfce 4 组件进行设置的途径。启动 Xfce 4 设置管理器的方法为选择“Application”|“设置”|“Xfce4 设置管理器”菜单命令，其操作主界面如图 9-20 所示。此对话框显示了 Xfce 4 所有组件的配置链接，在这里简述几个经典的配置。

9.5.1 设置显示器分辨率和刷新率

在 Xfce 桌面环境中设置显示器的具体操作步骤方法为：在设置对话框中，单击“显示”按钮，弹出“显示”对话框，然后选择分辨率下拉列表，选择“1024×768”选项，参数设置完毕如图 9-21 所示。单击“应用”按钮，完成操作，单击“关闭”按钮关闭“显示”对话框。

9.5.2 设置首选应用程序

对 Firefox 情有独钟的用户，可以将 Firefox 设置为 Xfce 桌面环境下的默认网页浏览器，具体的设置方法为：单击“设置”对话框中的“首选应用程序”按钮，弹出“首选应用程序”对话框，切换至“网络”选项卡，在网页浏览器下拉列表中选择“Mozilla Firefox”选项，如图 9-22 所示。设置完毕，单击“关闭”按钮关闭对话框。下次打开网页文件的时候，Xfce 桌面环境会调用 Firefox 应用程序打开网页文件。



图 9-20 Xfce 4 设置管理器

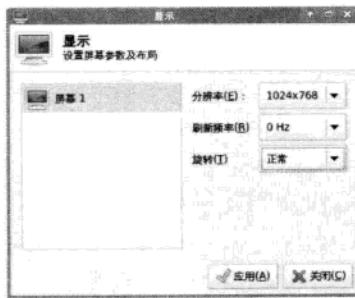


图 9-21 设置分辨率



图 9-22 设置首选应用程序

9.5.3 设置工作区桌面的名称

工作区桌面是 Xfce 4 桌面环境的一个重要的组成部分。Xfce 4 桌面环境允许用户对工作区桌面进行命令，以方便用户同时操作多个工作区桌面，具体的设置方法为：单击设置对话框中的“工作区”按钮，打开“工作区”对话框，在工作区桌面列表中，依次选择 1 和 2，并在输入栏中将原来的工作区名称分别改为“学习”和“娱乐”，如图 9-23 所示。

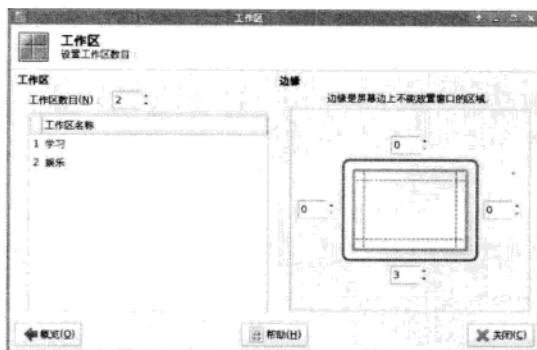


图 9-23 更改工作区名称

如此设置之后，用户将鼠标放置在底部面板的桌面切换器上第一个矩形框之上，系统会提示信息：单击切换到“学习”，这样就可以激活“学习”桌面。

补充说明

在工作区对话框中，工作区数目文本框的数字表示当前系统包括了多少个工作区桌面，单击后边的向上箭头按钮，工作区桌面的数目增加至 3，同时工作区列表中会增加多一条记录，序号为 3，工作区名称为“桌面 3”。若单击向下箭头按钮，工作区会减少一个，并且工作区列表中减少一条记录。

面板的管理与桌面的设置在上一节中已做简述，在这里就不详述，在设置对话框中单击“面板”按钮弹出图 9-14 所示的“面板”对话框；单击“桌面”按钮弹出图 9-16 所示的“桌面设置”对话框。

9.6 Xfce 的组件和常用应用程序



Xfce 桌面环境由许多组件组成。每个组件各司其职，互相协助，为用户提供了一个精简易用的 Xfce 桌面环境。Xubuntu 集成了许多基于 Xfce 桌面环境而开发的常用应用程序。

9.6.1 Xfce 4 的核心组件

Xfce 4 项目体现了 UNIX 哲学中的模块化和可复用性。Xfce 4 桌面环境作为图形化桌面环境，它包括了窗口、菜单、面板、图标和其他图形化元素的组件，它的各个组件之间既是



相互独立开发的，又是紧密联系的。

- 窗口管理器 Xfwm 4

管理应用程序窗口在屏幕上的布局，提供窗口风格并管理工作区，还提供了以快捷键操作的功能。

- 面板管理器 Xfce 4-panel

Xfce4 面板放置程序启动器、工作区切换器、时钟、菜单等。Xfce 4 面板程序方便用户操作，给用户一个美化的外观，而且面板还支持面板上的图标设置，支持尺寸的改变，以及面板数量上的增减。

- 桌面背景管理器 Xfdesktop

此程序用来设置背景图片和颜色，并且提供了一个根窗口菜单、菜单面板插件和菜单编辑器。

- 打印助手 Xfprint

Xfprint 提供打印图形前端的功能。Xfprint 可以使用 CUPS、BSD-LPR 或一个文件作为目的文件，还可以打印 Word 文件，图片文件等。

- 会话管理器 Xfce4-session

会话管理器控制着 Xfce 桌面环境的启动和关闭。注销 Xfce 时它能保存当前运行应用程序的状态（如果应用程序支持的话），并在你下次登录时正确载入这些应用程序。

- 设置管理器 Xfce4-setting

这是一个新的设置对话框，它包含了其他设置的链接。系统的设置文件，配置应用程序都可在这一对话框中找到相应的链接。

- Xfce 库

libxfcogui4、libxfc4mcs 和 libxfc4utils，这 3 个库几乎被所有 Xfce 程序用到，而且程序员在开发基于 Xfce 桌面环境的应用程序时也会使用这几个包。

- 实用工具与脚本

这一分类包含了必要的实用工具与脚本。向用户提供一个“关于”对话框，一个运行对话框，以及许多其他重要的被其他软件包使用的脚本，可使用 bash 和 perl 脚本语言。

9.6.2 应用程序

Xubuntu 中集成了 Xfce 的应用程序，同时也使用 Ubuntu 中的大部分性能强大的软件，并做相应的修改。以下介绍 Xubuntu 中几种常用的应用软件，用户也可以到 Xfce 4 的官方网站查看与 Xfce 4 项目有关的子项目，网址为：<http://www.xfce.org/>。

- Orage

Orage 是一款带提醒功能的简单日历程序。开启计算机时，Orage 会自动弹出，并显示用户设定的任务和约会记事。

用户可编写日志，记录自己的行踪，添加事件，设置提醒哪一天要做什么事情，设置提醒指定一天内要完成什么任务等。Orage 还提供了语音提示功能，通过播放语音，更有效地提醒用户。

Orage 是一个每天都需要用到的应用程序，这样 Xfce 4 桌面环境就把它作为一个守护进

程运行在系统的后台。

启动 Orage 应用程序的方法有多种，其中包括：单击“Application”|“办公”|“Orage”按钮；双击顶部面板的 Orage 启动器图标。

- Adiword 电子文档编辑器

Adiword 项目是一个开源的项目，它创建的目的就是创建一个稳定的，可兼容多个编辑平台的文字处理程序。目前，Adiword 完全免费，广泛应用于 Linux (X11)，Linux (GNOME)，Windows 95、98、Me、NT、2000 和 XP，还可以应用于 Max OS。

Adiword 的一个重要特点是使用了 xml 保存文件，这使得其他的文档处理程序或文本编辑器可方便地读取 Adiword 保存的文档。

启动 Adiword 文档编辑器的方法有多种，其中从应用程序菜单启动的方法为：选择“Application”|“办公”|“Adiword”菜单命令。

- Mousepad 文本编辑器

Mousepad 是用于 Xfce 的一个简单的文本编辑器，其基于 Leafpad。最初创建它的动机是想提供由于多种原因在 Leafpad 上难以实现的打印支持。

Mousepad 操作简单，支持多种编码格式，如 Unicode、Utf-8、GB2312。用户可以同使用 NotePad 一样编辑文本。

启动 Mousepad 包括两种方法：选择“Application”|“附件”|“Mousepad”菜单命令和双击文本文件。

- Xfce 4Terminal

Xfce 4 Terminal 是用于 Xfce 4 的一个轻量级终端模拟器。它是现代、轻量级、内存占用低、具有标签和多窗口操作功能的终端，提供键绑定、外观、颜色等的完全设置，可复制代码，粘贴代码。

启动 Xfce 4 模拟终端的方法为：选择“Application”|“附件”|“终端”菜单命令。

- Thunar 文件管理器

Thunar 是 Xfce 4 桌面环境的一个新的、现代的文件管理器。为符合 Xfce 的设计理念，Thunar 的界面设计得相当简洁，设计的目标就是运行快速，内存占用少。Thunar 作为文件管理器，它实现了创建和移动目录、管理文件和访问移动介质等功能。

只要打开文件夹就可启动 Thunar 应用程序。

- 图像查看器

Xfce 4 中的默认图像查看器为 Ristretto，是一个轻巧的图像查看器。

- Dictionary 字典

Dictionary 是一个日常使用的英文翻译软件。Dictionary 支持将英语翻译为中文、德文、西班牙文等功能，而且具有英汉解释，并有相应的例子作为辅助解释，可设置从本地磁盘上搜索词汇，也可指定从网上搜索。Dictionary 除了可以搜索到词汇的内容解释，还可以查询词汇的发音。



9.7 小结

本章简单介绍了 Xfce 桌面系统，阐述了 Xfce 和 Xubuntu 的发展历史，介绍了 Xfce 的核心组件，包括窗口管理器、面板管理器、会话管理器、设置管理器和打印助手等。Xubuntu 中集成了 Xfce 高效的常用应用程序，包括 Orage 简单日历程序、Adiword 电子文档编辑器和 Mousepad 文本编辑器等。

Ubuntu 发行版本系列中，Ubuntu 使用 GNOME 桌面系统作为默认的桌面环境，Kubuntu 使用的是 KDE 桌面环境，Xubuntu 则使用了 Xfce 桌面环境作为它的默认桌面环境。3 种桌面环境各有各的特点，GNOME 和 Xfce 都是基于 GTK+ 开发包的，而 KDE 则是基于 Qt 开发包开发的。Xfce 适合配置比较低的计算机，运行速度快。3 种桌面环境都是可设置的。对普通用户来说，使用 Ubuntu 比较好。Ubuntu 操作简单，功能强大，所有配置都有相应的图形化操作界面，包括所有日常生活、工作中使用的软件以满足大众的要求。



9.8 习题

1. 在 Ubuntu 中使用新立得软件包管理器安装 Xubuntu。
2. Xfce 的起源？Xfce 开始时使用什么开发包？
3. Xfce 的核心组件有哪些？
4. 在 Xfce 桌面环境下，添加一个新面板，并向此面板添加新项目，调整新面板的位置大小。
5. 修改 Xfce 桌面的背景图片。
6. 设置在 Xfce 桌面上不显示默认图标回收站。
7. 打开 Mousepad 应用程序，观察 Mousepad 与 Windows 下的记事本有什么不一样。



Linux

第4部分 基本应用

第10章

shell 基础应用

Ubuntu 提供了简单易用的图形化操作界面给用户，同时保留了 Linux 下功能强大的字符操作界面。shell 命令的使用能够提高使用和管理 Linux 的效率。

Ubuntu 系统下的 shell 命令使用十分广泛，学习 Ubuntu 也就必然需要学习 shell 命令的使用方法。本章主要介绍了 shell 的基础概念，讲述了 shell 命令行的语法和使用技巧。本章的结尾将通过实例讲解 Ubuntu 中常用的操作命令，让读者更深入了解 shell 命令的使用。





10.1 认识 shell

shell 的中文意义为外壳，如同 Linux 的外壳保护着 Linux 的内核，并且作为与外界通信的桥梁。shell 是命令解释器同时也是一种高级编程语言。作为命令解释器，它解释和执行用户输入的命令；作为高级编程语言，它能够编写出可运行的脚本，提供函数和控制结构等语言编程能力。

10.1.1 shell的概念

shell 外壳主要是为了区别于 Kernel 内核，它是一个用户的操作接口，负责接收用户的命令输入，解释命令并调用相应的应用程序执行指定的操作。其实，shell 的概念并不是在 UNIX/Linux 中才有的，这一概念在 Windows 系列中也有，只不过其表现为 DOS 的 COMMAND.COM 命令解释器或后来的 cmd.exe。

shell 按是否使用图形界面作为接口的方式分类，可分为两种：图形界面的 shell 和字符界面的 shell。图形界面的 shell 即 GUI shell，提供了一个图形的操作界面，其中包括我们在 Linux 下所使用的 GNOME、KDE 和 Xfce 桌面环境，以及在 Windows 下使用的 Windows Explorer。字符界面的 shell 即 CLI shell，提供一个字符型的操作界面，其中使用比较广泛的 CLI shell 包括 DOS 下使用的 COMMAND.COM、Windows XP 下使用的 cmd.exe 及 UNIX/Linux 下使用的 bash。

目前，通常所说的 shell 都是指 UNIX/Linux 下使用的 CLI shell。它作为命令解释器解析用户输入的程序命令，在系统中查找相应的应用程序，然后向 Linux 内核发出操作请求，并将系统的控制权交给内核，由内核负责执行应用程序，而内核完成执行任务后将操作结果传递给 shell，并交出控制权给 shell，shell 回应用户的操作。若用户输入的命令为 shell 的内置命令，shell 自行执行操作，不需要内核的参与，这样 shell 内置命令的执行时间短，效率高。

shell，作为命令解释器，其解释的命令一般可分为程序命令和 shell 内置命令。程序命令表示保存在磁盘中可运行的文件命令；而内置命令表示命令不是一个可执行的文件而是可由 shell 直接解释执行的命令字符串。

如何区分程序命令和内置命令呢？使用 whereis 查看所输入的命令是否有一个与之相应的可运行文件保存在磁盘中。若所输入的命令是一个可执行的命令，并且在使用 whereis 命令查看此输入命令的相关信息时并在磁盘中找到与之相对应的可运行文件，那么此命令为 shell 内置命令。例如 history 命令，在终端输入 history 命令并按下回车键，我们可以看到如下结果：

```
sunny@sunny-desktop:/ $ history 3
503 history
504 history 5
505 history 3
```

这表示 history 命令是一个可执行的命令，然后使用 whereis 命令查看 history 命令的位置，输入命令并按下回车键，执行结果如下：

```
sunny@sunny-desktop:/usr$ whereis history
history: /usr/share/man/man3/history.3readline.gz
```

从代码中可以看到并没有一个可运行的文件与 history 命令相对应。命令 whereis 的执行结果中只显示了 history 帮助文档的路径，由此说明 history 是 shell 的一个内置命令。

而我们通常使用的 cp 命令则是一个程序命令，它的可执行文件在/bin 目录下，执行命令 whereis 的显示结果如下所示：

```
sunny@sunny-desktop:/usr$ whereis cp
cp: /bin/cp /usr/share/man/man1/cp.1.gz
```

10.1.2 shell 的类型

shell 产生于 UNIX 操作系统，随着 UNIX 和 Linux 的发展，shell 的类型越来越多。Ubuntu Linux 支持的 shell 都记录在/etc/shells 文件中，用户可使用 cat 命令来查看此文件的内容。执行命令 cat，执行结果部分显示如下所示：

```
/ $ cat /etc/shells
# /etc/shells: valid login shells
/bin/csh
/usr/bin/es
/usr/bin/ksh
.....
/bin/bash
/bin/rbash
/bin/false
```

目前 UNIX/Linux 操作系统兼容的 shell 类型比较多，其中较为流行的是 Bash、Csh 和 Ksh。

- Bash

Bash 是 Bourne Again Shell 的缩写，它基于一个早期的 UNIX shell 即 Bourne shell。Bourne shell 是由 AT&T 贝尔实验室的 Stephen Bourne 编写的，它的执行效率要高于当时其他的 shell，所以它是大多数商业 UNIX 操作的基本 shell，但它不支持别名与历史记录功能。

Bash 于 1987 年由 Brian Fox 所创建，是 GNU 操作系统的 shell，后来成为多数类 UNIX 操作系统的默认 shell，Ubuntu 就是使用 bash 作为它的默认 shell。因为 Bash 基于 Bourne shell (sh)，所以 Bash 和 sh 有很好的兼容性，大多数 sh 脚本可以在不修改的情况下在 Bash 中运行。在 Linux 中 sh 是 Bash 的一个链接，当使用 sh 执行命令时 Bash 会调用模拟的 sh 环境执行命令。

Bash 的命令别名、命令行编辑、历史记录特性来源于 Ksh 和 Csh 的类似特性。Bash 的这些改进弥补了 sh 的不足。

- Csh

Csh 是 UNIX shell 的一种，全称为 C Shell，是由 Bill Joy 在美国加利福利亚大学伯克利分校为 BSD 系统开发的，PX 的名称缘于其语法，类似于 C 语言语法。Csh 基于第 6 版本 UNIX 系统中/bin/sh，并且在此基础上新增了许多特性，包括别名、命令历史和文件名替换、作业控制等功能。

目前 Csh 并没有被广泛地使用，尽管它的交互性能好，可是它的脚本执行能力却受到不少的批评。在 1980 年左右 Kern Greer 为 TENEX 开源操作系统编写 C Shell 的时候，并对 C Shell

出所有匹配的项目。这一特征帮助用户快速输入系统命令，同时降低用户记忆命令的难度。

Ubuntu 中 Bash 的自动补全功能可补全的对象包括环境变量、文件名、文件夹名和命令等。

- 别名功能

在终端启动 shell 之后，为了方便用户使用常用的命令，Bash 提供了命令别名的功能，即用户可使用 alias 命令自定义一个简单易记的别名命令代替复杂的命令，或代替多命令组合的命令行。

- 后台处理

一般来说，在 Ubuntu 中 shell 作为一个交互式模式的命令解释器，它需执行完一个命令之后才能执行下一个命令，可在 Bash 中，Bash 允许多个命令同时执行，就是将一些程序放置到后台去执行。

通常我们会将比较耗时的工作放在后台执行，如编译源代码、从网络下载软件包的工作和使用 apt-get 命令安装软件包等，在后台执行这些命令的同时 shell 还可以接受其他命令的输入。

- 通配符功能

Bash 还有一个重要的功能就是可使用通配符。使用 Bash 中的通配符功能可以快速找到自己所需要的文件和文档。在一个包括很多文件的文件夹里若要查看一个文件，单靠自己的眼睛去看是很费劲的，使用 Bash 中的通配符能够让用户快速定位到所指定的文件。

10.1.4 命令行语法

终端就是 Bash 的运行环境。打开终端就启动了 Bash shell。在终端屏幕上的命令提示符后输入命令并按下回车键后，Bash 就会按照一定规则去执行命令。命令行的语法说明了行中每个元素的排列顺序和间隔方式。Bash 解释命令的规则就是 shell 的语法。命令行上命令的基本语法格式如下所示：

```
command [arg1] [arg2] ... [argN]
```

输入命令之后按回车键，执行命令。命令行采用空格来分隔每个元素。其中 command 表示命令名，arg1 到 argN 表示命令的参数，参数可以是操作的目标或是命令的选项，使用中括号将参数括起来表示这些参数是可选项。

- 命令名

命令名标识了命令的作用。在 Linux 直接输入命令名时，若该命令可以执行不带参数的命令行，它能够成功地被执行，并显示执行结果。例如 ls 命令不带参数表示列出当前文件夹下的文件列表。若该命令不可以执行不带参数的命令行，那么 shell 会提示用户此命名的用法信息或提示“此命令不可用信息”。例如 cp 命令，它是一个复制文件的命令，它必须有命令参数，而当用户只输入 cp 命令的时候，显示结果如下：

```
sunny@sunny-desktop:/$ cp  
cp: 缺少了文件操作数  
请尝试执行 “cp --help” 来获取更多信息。
```

参考错误信息，用户可以输入 cp --help 获得 cp 命令的使用方法。

- 参数

在 shell 中，参数表示命令行上每一串不含空格字符的字符串，它只是一种对命令行位置



的标识。命令行上的参数都是有编号的，且编号都是从 0 开始。参数 0 表示命令名，它是命令参数的开始。而通常来说，参数是除命令名外的其他字符串，它可以是文件名、路径、命令的选项，字符串等。

下面的代码将显示目录/home/sunny 的详细文件列表的命令行：

```
sunny@sunny-desktop:~/ls /home/sunny/ -l
```

ls 是命令名，其参数编号为 0。参数/home/sunny/ 为文件目录，其参数编号为 1。参数-l 为 ls 命令的选项，其参数编号为 2。而没有带参数的 cp 命令是不能被执行的，也就是说 cp 命令在创建的时候，就要求必须要有参数才可以运行，其实 cp 要求至少有两个参数才可以被执行。

- 选项

选项是改变命令执行效果的参数。通过使用多个选项，命令可以按照不同的方式执行，并能产生不同的执行结果。不同的命令会拥有不同的命令选项，命令的选项在命令被编写完成的时候就已被定制。对于程序命令来说，命令的选项解释并不是由 shell 来完成的，shell 作为一个命令解释器，它调用命令并将命令后的选项传递给命令本身，让命令解释命令的选项。

通常选项的名称是完整的英文字符串，而选项前面都会有两个连接符号，如--help 选项；如果选项为一个字符，选项前面会有一个连接符号，如-f。有时候会出现这样一种情况就是-f 和 --force 选项表示同一个意思，也就是说两个选项同时表示同一种执行效果，-f 为--force 的缩写。

选项是可以合并的。如果你输入 ls -a -l 命令，那是多么费劲的一件事情。Bash 允许用户合并选项，即命令 ls -a -l 和命令 ls -al 是等同的。

选项出现的顺序是无关紧要的，即命令 ls -al 和命令 ls -la 是等同的。

命令的选项还可以附带选项的参数，因有些工具的选项是需要参数的。如工具 gcc 的-o 选项后必须跟 gcc 产生的可执行文件名，通常选项与其参数间用空格分开如下所示：

```
sunny@sunny-desktop:~/doc$ gcc -o main main.c
```

有时为了区分使用“-”连接字符开头的文件和选项，一般在输入命令行的时候，需要在参数的前面添加两个连接字符表示，以一个连接字符开头的参数开始，如下所示：

```
sunny@sunny-desktop:~/doc$ cat -- -test
```

此时--test 为参数，而不是选项。若在使用 shell 命令的时候遇到难题，不妨使用--help 选项。

10.2 标准输入输出



在 shell 终端里，命令并不知道需要处理的源信息从哪里来，也不知道输出的信息要到哪里去，shell 终端管理着命令的输入和输出。

shell 从键盘上获得信息，并将输入传递给命令，shell 将命令的执行结果输出到屏幕，也可以输出重定向到其他文件。shell 的标准输入为键盘和文件，而标准输出为屏幕和文件。在 shell 中除了标准输入和标准输出，还有一种错误信息的输出，就是标准错误输出。

在默认情况下，shell 的标准输入是键盘，而默认的标准输出是屏幕。使用 cat 命令来讲述 shell 默认情况的标准输入和标准输出。

命令 `cat` 是一个把键盘作为标准输入，而把屏幕作为标准输出的命令。命令操作过程为在命令提示符后输入 `cat` 命令按下回车键，然后从键盘上输入 `this is test1` 字符串后按下回车键，此时屏幕会在输入行下面显示 `this is test1`，再次输入 `this is test 2` 字符串后按下回车键，此时在输入行的下面显示 `this is test 2`，如此反复输入，输入的字符串都会在下一行显示。执行此过程的代码与执行结果如下所示：

```
$ cat
this is test1
this is test1
this is test2
this is test2
```

按下 `Ctrl+C` 组合键，结束 `cat` 命令的执行。按下 `Ctrl+C` 组合键向 `cat` 发送 EOF 信号，该信号表明标准输入结束，那么 `cat` 就停止执行，并将控制权返回给 `shell`，`shell` 恢复等待命令输入状态。此过程表明，`cat` 命令从标准输入中获得了信息，并将信息显示到标准输出（屏幕）。

10.2.1 重定向

重定向是指改变 `shell` 标准输入和标准输出的方式。如前所述，在默认情况下，`cat` 的标准输入是键盘，而其标准输出是屏幕。使用重定向，命令的信息输入可以来源于文件，信息输出可以重定向到文件。本节将介绍如何将输入、输出重定向到普通文本。

1. 重定向标准输入

为了使命令使用的数据来自磁盘上的文件，`shell` 提供了重定向标准输入的功能。使用重定向符号（<）可以将 `shell` 命令的输入重定向为指定的文件而不是键盘。这种重定向标准输入的命令格式为：

```
command [arguments] < filename
```

其中 `command` 表示命令名，`arguments` 为参数列表，<为重定向符号，`filename` 表示普通文件的文件名。

例 10.1：命令 `cat` 的标准输入被重定向

假设在主目录下已有一个普通文件名为 `simple`，此 `simple` 的内容为“`This is demo`”文本，`shell` 的当前路径为`~`，那么执行 `cat < simple` 命令之后结果如下所示：

```
~$ cat < sample
This is a demo!
```

字符串“`This is a demo!`”是从 `sample` 文件中读取出来的。此时，在屏幕上显示了这一字符串，表明 `cat` 从 `sample` 中获得了信息输入。系统会在普通文件结束时自动产生 EOF 信号以结束文件读取，`cat` 接收到信号之后，结束执行。

`shell` 中还存在另一种重定向成为 `here` 文档。重定向符号“`<<`”，它结合一对分隔符，其功能是将键盘上输入的文本重定向到命令。分隔符可以是其他任意字符或字符串，这对分隔符一定是相同的字符或字符串组成，而且第一个分隔符要紧跟在重定向符号“`<<`”的后面，第二个分隔符一定要自成一行。

例 10.2：命令 `cat` 的 `here` 文档重定向

```
~$ cat << man
> man cat
> man ls
```



```
> man bash  
> man  
man cat  
man ls  
man bash
```

代码中的第一行为命令，第 2~5 行为键盘上输入的字符串，在第 5 行 “> man” 输入后按下回车键，第 2~4 行的字符串显示在第 6~8 行上。此处字符串 man 作为分隔符。

由于大多数命令都是以参数的形式在命令上指定输入文件的文件名，所以重定向输入并不是经常使用。如 cat filename 与 cat < filename 的执行结果是一样的。接受文件名作为参数的命令工具还包括 lpr、sort 和 grep。对一些不接受文件名作为参数的命令，使用重定向输入是很有必要的。

2. 重定向标准输出

重定向标准输出是指 shell 通过使用重定向符号 “>” 将 shell 命令的输出重定向到指定的文件而不是屏幕上。这种重定向标准输出的格式为：

```
command [arguments] > filename
```

其中 command 表示命令名，arguments 表示参数列表，“>” 表示重定向符号，filename 表示指定的重定向普通文件。

例 10.3：命令 cat 的标准输出被重定向

假设在主文件夹下已保存了一个 sample 文件，此文件的内容为 “This is demo!”。使用 cat 读取此文件，并将 sample 中的内容输出重定向到 simple1 文件中。然后使用命令 cat 查看 simple1 文件的内容。命令执行过程和执行结果如下所示：

```
~$ cat sample >sample1  
~$ cat sample1  
This is a demo!
```

在默认情况下，若 sample1 文件不存在，那么 sample1 文件会自动被创建；若存在，那么 sample1 文件被重写。因此，使用 cat 的重定向输出的时候需要特别注意。shell 提供了一种防止重定向时不经意地重写了已存在文件的功能，此功能称为 noclobber 功能，可以使用 set -o noclobber 命令启动 noclobber 功能，使用 set +o noclobber 命令关闭 noclobber 功能，在默认情况下，noclobber 功能是关闭的。开启 noclobber 功能，执行重定向输出命令，再关闭 noclobber 功能，命令执行过程和执行结果如下所示：

```
~$ set -o noclobber  
~$ cat sample > sample1  
bash: sample1: 无法覆盖已经存在的文件  
~$ set +o noclobber  
~$ cat sample > sample1
```

而使用重定向符号 “>>” 可以将标准输出重定向追加到指定文件中。

例 10.4：使用命令 cat 将 sample 文件的内容追加到 sample1 文件

由例 10.3 可知，sample 和 sample1 文件的内容都拥有一行 “This is a demo!” 字符串，使用命令 cat 将 sample 中的内容追加到 sample1 里，然后查看 sample1 文件的内容。命令执行过程和执行结果如下所示：

```
~$ cat sample >>sample1  
~$ cat sample1  
This is a demo!
```

This is a demo!

同标准输出使用重定向符号“>”和“>>”重定向输出一样，标准错误输出使用重定向符号“2>”和“2>>”重定向输出。一般来说，命令执行没有错误是不会有标准错误输出的，唯有执行错误的命令，shell 才会输出标准错误输出。

例 10.5：命令 cat 的重定向标准错误输出

执行错误的命令 `cat -d sample`，将错误信息输出重定向到 `error` 文件中，然后查看 `error` 文件的内容，命令执行过程和执行结果如下所示：

```
~$ cat -d sample 2>error
~$ cat error
cat: invalid option -- 'd'
请尝试执行 "cat --help" 来获取更多信息
```

例 10.6：命令 cat 的重定向标准错误输出追加到文件

由例 10.5 可知，文件 `error` 已包含了两行的错误信息，现在使用重定向符号“2>”，再次执行错误命令，然后查看 `error` 文件的内容，执行过程和执行结果如下所示：

```
~$ cat -d sample 2>>error
~$ cat error
cat: invalid option -- 'd'
请尝试执行 "cat --help" 来获取更多信息
cat: invalid option -- 'd'
请尝试执行 "cat --help" 来获取更多信息
```

补充说明

以上显示为“`~$`”开头的行，表示命令行，`$` 表示命令提示符，其前面的`~`符号表示当前的工作目录为用户的主文件夹。命令提示符“`$`”前表示的就是工作路径，后面的字符串表示命令。

系统的标准输入、标准输出和标准错误输出是分别使用 0、1 和 2 表示的。因此，在重定向标准错误输出时使用了“2>”的重定向符号。

10.2.2 管道

shell 使用管道将一个命令的输出直接连接到另一个命令的输入。若能对管道操作进行分解，管道操作的执行过程相当于如下过程：首先使用一个命令的重定向输出到一个辅助文件中，另一个命令则使用重定向输入此辅助文件，格式如下所示：

```
command1 [arguments]> temp
command2 [arguments]< temp
```

这样的执行效果不是很好，而且还需要辅助文件来协助才能达到在命令之间传递数据。使用管道，shell 不会单独执行一条命令，而是将几条命令作为一条命令来执行，执行效率高，而且不需要辅助文件。管道的符号为一条竖线“|”，命令格式如下所示：

```
command1 [arguments] | command2 [arguments]
```

例 10.7：对文件 sample 进行排序后重定向到 sample1 文件

假设在`~/doc` 目录下已有 `sample` 文件，现将 `sample` 的内容进行排序，然后将排序后的结果保存到 `sample1` 文件中，命令执行过程和执行结果如下：

```
~/doc$ cat sample
hello 4
```



```
hello 2
hello 5
hello 3
hello 1
~/doc$ cat sample | sort >sample1
~/doc$ cat sample1
hello 1
hello 2
hello 3
hello 4
hello 5
```

例 10.8：将命令 ls 的输出通过管道传递给 less 或 more 命令

```
$ ls | more
```

例 10.9：在/etc/passwd 文件中查找记录 root 用户的记录行
命令执行和执行结果如下所示：

```
$ cat /etc/passwd | grep root
root:x:0:0:root:/root:/bin/bash
```



10.3 通配符

使用以下特殊的字符代替部分文件名，shell 可以自动生成与已有文件相匹配的文件名。而这些特殊的字符就是通配符。使用通配符，能够帮助用户模糊查找到所需操作的文件。含有通配符的字符串称之为模式。

10.3.1 通配符？

通配符?表示匹配任意一个字符，可与其他字符串组成一个模糊的文件名，用以匹配文件。为了说明通配符的使用，我们需要做如下步骤：

(1) 在主文件夹创建一个空的文件，使用命令如下所示：

```
~$ mkdir test
```

(2) 接着进入 test 目录，使用 touch 命令，创建多个空文件，执行命令如下所示：

```
~$ cd test
```

```
~/test$ touch hello1 hello2 hello13 hil hi2 hi3 hal ha2
```

(3) 使用 hello? 模式字符串可以配 hello1 和 hello2，但不可以匹配 hello3。

例 10.10：通配符?放置在模式字符串的后面

使用 ls 命令显示与 hello?字符串匹配的文件，执行命令和执行结果如下所示：

```
~/test$ ls hello?
```

```
hello1 hello2
```

例 10.11：通配符?放置在模式字符串的中间

使用命令 ls 显示与 h?2 模式字符串的文件，执行命令和执行结果如下所示：

```
~/test$ ls h?2
```

```
ha2 hi2
```

使用通配符?可以比较精确地匹配文件名，方便在一个包含多文件的目录下寻找指定的文件。

10.3.2 通配符*

通配符*的功能与通配符?的功能相当，只是通配符*的匹配范围更加广泛，它可以匹配任意多个字符而不是一个。接着使用上小节的示例文件，此时工作目录为主文件夹下的 test 文件夹，里边包括了一定数量的文件。

例 10.12：通配符*位于字符串的中间

使用模式 h*1 来查找文件，执行命令并显示执行结果如下所示：

```
~/test$ ls h*1
hal hello1 hil
```

发现执行结果显示的文件中，第一个字符为 h，结尾字符为 1。与通配符*匹配的字符对应的有 a、ello 和 i。

若用于复制多个文件命令，通配符*的使用可以简化文件的输入。

例 10.13：使用通配符*复制文件到指定目录

将 test 目录下的以 hi 开头的所有文件复制到主文件夹的 document 文件夹中，只需执行如下命令：

```
~/test$ cp hi* ~/document
```

若直接使用*作为模式的话，表示匹配文件夹中所有显性的文件和文件夹（不包括隐藏的文件和文件夹）。例如若要删除 test 文件夹中的所有文件，就可以使用 rm *命令。

补充说明

一般来说，Linux 下的隐藏文件或文件夹都是以“.”开头的。文件管理器在默认的情况下不会将隐藏文件或文件夹显示出来。直接使用命令 ls 也不会将它们显示出来，但使用 ls -a 可将当前目录下的文件罗列出来，其中就包含隐藏的文件或文件夹。

10.3.3 字符范围符[]

字符范围符[]可以匹配任意一个在中括号内的字符。这种方式匹配更有针对性。默认 hello[13]只能匹配到 hello1 和 hello3 文件，其他的文件是不能被匹配的。

例 10.14：假设~/test 目录中有以下文件 pic1.jpg、pica.jpg、pic3.jpg、picb.jpg、pic5.jpg、picd.jpg、pice.jpg 和 picf.jpg，现在需要将前面包含 pic 字符且带有数字的.jpg 格式文件复制到主文件夹下一个空的 pic 目录里边，执行命令如下所示：

```
~/test$ cp pic[12345].jpg ~/pic
```

执行命令后使用命令 ls 查看~/pic 目录上是否已有文件，命令执行与执行结果如下所示：

```
~/test$ ls ~/pic
pic1.jpg pic3.jpg pic5.jpg
```

pic[12345].jpg 模式中，字符范围符[]中包括了 5 个字符，这样的输入匹配不是很高效，shell 允许用户使用连字符“-”表示数字或字母的范围。pic[12345].jpg 模式与 pic[1-5]模式是等价的。

字符范围符[]除了用来限定数字，还可限定英文字母。[a-z]就代表英文字符中所有的小写字母，[a-zA-Z]代表了所有英文字符中的大小写字母。

通配符?、通配符*和字符范围符[]之间是可以组合在一起使用的。若 p?c[1-5].*表示可以



匹配的以 p 字母开头的，p 字母下一个字符为任意字符，任意字符后接着一个 c 字母，紧跟其后的是数字 1 到 5 中的其中之一，然后接着是一个“.”字符，最后*字符表示匹配任意字符。这个模式可以匹配的文件名包括：pic1.jpg、pac2.txt、pfc3.pencil 和 pdc5.a 等。

技术提示

使用字符范围符[]时还可在左方框后紧跟着逻辑非符号“!”或脱字符号“^”，表示不匹配[]里边的字符。如[!ab]表示不匹配 a 和 b 字符；[^sb]*表示不匹配以 s 和 b 开头的文件名。

10.4 shell 操作



在 Ubuntu 中 shell 的命令行使用是必不可少的。文件的处理与管理，系统服务的管理，系统故障的维护等都可以在 shell 的模拟终端中轻松完成。在 Ubuntu 中使用的模拟终端是 GNOME-terminal 应用程序。本节将讲述在 GNOME-terminal 上如何运行 shell 命令。

10.4.1 启动终端

模拟终端是执行 shell 命令的一个环境。Ubuntu 的模拟终端——GNOME-terminal 命令保存在 /usr/bin 目录下。启动模拟终端的方法有两种：选择“应用程序”|“附件”|“终端”菜单命令；按下“Ctrl+F2”组合键，弹出“运行应用程序”对话框，在顶行的文本框中输入字符串 GNOME-terminal，如图 10-1 所示，然后单击“运行”按钮。

启动模拟终端 GNOME-terminal 后，进入运行界面如图 10-2 所示。



图 10-1 运行 GNOME-terminal 应用程序



图 10-2 GNOME-terminal 主界面

只要在命令提示符后面输入命令，然后按下回车键，就可执行命令。在提示符后输入 echo \$SHELL 命令，查看终端使用何种 shell，执行命令和执行结果如下所示：

```
dranyu@dranyu-desktop:~$ echo $SHELL  
/bin/bash
```

从上述执行结果，可知道当前使用的 shell 是 Bash。

补充说明

图中第一行字符串“dranyu@dranyu-desktop:~\$”是提示符，其中 dranyu 是登录当前会话的用户名，符号@表示“在”的意思，而 dranyu-desktop 是指主机名，符号~是指当前工作目录为用户 dranyu 的主目录 (/home/dranyu)，符号\$为普通用户的命令提示符。若登录 shell 的用户为 root 用户，命令行提示符号为#，因为 root 是系统管理员。普通用户可通过 sudo 命令获得系统管理员的权限。具体如何设置命令行提示符在 10.4.3 节中讲述。

10.4.2 更换 shell

Ubuntu 支持多种 shell，这些 shell 都可以运行于终端上。Ubuntu 终端默认的登录 shell 是 Bash。用户可以根据个人的使用习惯选择使用不同的 shell。此时就更换 shell。更换的方法很简单，只要在终端上执行相应的 shell 程序名称，即可完成 shell 的切换。若系统没有安装此 shell，终端会自动提示此 shell 没有安装，并提供了安装的方法。

在终端执行 Csh、Tcsh、Ksh 和 sh，执行结果如下所示：

```
dranyu@dranyu-desktop:~$ csh
程序 'csh' 已包含在以下软件包中:
* csh
* tcsh
试试: sudo apt-get install <选定的软件包>
bash: csh: 找不到命令
dranyu@dranyu-desktop:~$ tcsh
程序 'tcsh' 尚未安装。 您可以通过输入以下命令安装:
sudo apt-get install tcsh
bash: tcsh: 找不到命令
dranyu@dranyu-desktop:~$ ksh
程序 'ksh' 已包含在以下软件包中:
* pdksh
* zsh
* ksh
* mksh
试试: sudo apt-get install <选定的软件包>
bash: ksh: 找不到命令
dranyu@dranyu-desktop:~$ sh
$
```

从显示结果看，Ubuntu 默认情况下并没有安装 Csh、Tcsh 和 Ksh。sh 是默认已经安装的。在 Bash 中运行其他类型的 shell，需要退出 shell 时，在命令提示符后输入 exit 命令即可。

终端上的 shell 是可嵌套的。若执行了多个 shell，退出模拟终端的时候，就需要执行多次 exit 命令，如下所示：

```
1 dranyu@dranyu-desktop:~$ sh //处于 Bash 命令解释器下，并且执行 sh 命令
2 $ tcsh //处于 sh 命令解释器下，并且执行 Tcsh 命令
3 dranyu-desktop:~> csh //处于 Tcsh 命令解释器下，并且执行 Csh 命令
4 dranyu-desktop:~> ksh //处于 Csh 命令解释器下，并且执行 Ksh 命令
5 $ bash //处于 Ksh 命令解释器下，并且执行 Bash 命令
6 dranyu@dranyu-desktop:~$ exit //处于第二个 Bash 命令解释器下，执行 exit 命令
7 exit //退出 Bash 时，在屏幕上显示的 exit 字符串
8 $ exit //返回到 Ksh 命令解释器下，与第 5 行对应，并且执行 exit 命令
9 dranyu-desktop:~> exit //返回到 csh 命令解释器下，与第 4 行对应，并且执行 exit 命令
```



```
10 exit
11 dranyu@dranyu-desktop:~> exit //返回到 Tcsh 命令解释器下，与第 3 行对应，并且执行 exit 命令
12 exit          //退出 Tcsh 时，在屏幕上显示的 exit 字符串
13 $ exit        //返回到 sh 命令解释器下，与第 2 行对应，并且执行 exit 命令
14 dranyu@dranyu-desktop:~$ //返回到登录的 Bash 命令解释器下
```

为了解说注释，在代码的前面添加了行号，在每行的结尾添加注释，注释以符号“//”开头。命令执行过程是先执行 sh、Tcsh、Csh、Ksh 和 Bash，然后再执行 exit 命令，依次退出 Bash、Ksh、Csh、Tcsh 和 sh，最后返回到默认登录的 shell 中。

以上 shell 之间的变更仅限于此次登录 shell，即此模拟终端，当此终端退出，所有的 shell 都退出。下次再启动模拟终端时，恢复为默认的 shell——Bash。

如何才能在启动的时候使用指定的 shell 呢？Ubuntu 中提供了一个 Csh (change shell) 的命令，用来修改登录 shell。将默认的登录 shell (bash) 修改 Ksh 的示例如下：

```
dranyu@dranyu-desktop:~$ chsh
口令：
正在更改 dranyu 的 shell
请输入新值，或直接敲回车键以使用默认值
登录 shell [/bin/bash]: /bin/Ksh
```

执行 CHSh 之后，终端要求用户输入口令，在“口令”行中输入用户的密码后按下回车键，接着提示输入指定的登录 shell，输入“/bin/Ksh”字符串，然后按下回车键，完成修改登录 shell 的操作。退出 Ubuntu 系统，重新登录系统，再启动模拟终端的时候就使用 Ksh 作为默认的登录 shell 了。

技术提 术

输入用户的口令是为了获得用户的认证。使用 Csh 命令修改登录 shell，只是修改当前用户的登录 shell，其他用户的登录 shell 并没有修改。若当前用户是 root，那么 root 的登录 shell 将被修改。其他用户的登录 shell 保持不变。由于在登录 shell 中可方便更换 shell，用户可随时执行不同的 shell 脚本。

10.4.3 配置 shell

配置 shell 能够提高用户的工作效率。启动模拟终端后，屏幕将显示提示符，shell 处于等待用户输入命令的状态。shell 的配置文件能够设定提示符的组合，可设置命令的别名，以及设置 shell 的环境变量。

1. shell 的初始化

在 Ubuntu 中，以下几个系统的设置文件影响着 shell 的特性和环境变量。

- **/etc/profile:** 在系统登录时，此文件是操作系统设置用户信息环境时执行的第一个文件，其为系统的每个用户设置环境变量信息。
- **/etc/bash.bashrc:** 此文件是每个用户运行 Bash shell 的时候都要执行的配置文件，其设定了一部分的环境变量和命令行提示符。因此此文件的配置信息影响着系统每个用户的 shell 环境。
- **~/.profile:** 此文件是在启动模拟终端时，用户使用其输入专属于自己使用的 shell 信息，当用户登录时，该文件仅仅执行一次。在默认情况下，其设置一些环境变量，

执行用户的`~/.bashrc`文件，如将主文件夹下的`bin`文件添加到`PATH`环境变量中。

- `~/.bashrc`: 此文件包含专用于当前用户的 Bash shell 信息，当启动模拟终端时以及每次运行新的 shell 时，此文件都会被读取。用户可在此文件中设置自己的 shell 变量。
- `/etc/environment`: 此文件定义了系统的环境变量。在默认情况下，只有`PATH`变量，即：`PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games"`。

若要修改`/etc/profile`、`/etc/bash.bashrc`和`/etc/environment`，用户必须拥有 root 管理员的权限。用户可自行修改`~/.profile`和`~/.bashrc`文件，以满足自己的需要。

下面将介绍如何设置命令行提示符、设置命令别名，以及添加环境变量。

2. 设置提示符

命令行提示符由一系列字符组成，每次在 shell 接受用户命令输入的时候都会出现在屏幕上。`PS1`环境变量设置命令行提示符的显示信息。此处讲述如何设置`~/.bashrc`文件中的`PS1`环境变量。启动模拟终端后，使用 Gedit `~/bashrc &`命令可打开`~/bashrc`文件。`PS1`环境变量的设置代码如下所示：

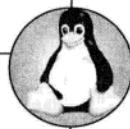
```
PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$'
```

其中``${debian_chroot:+ ($debian_chroot) }`是变量，`\u`代表当前登录的用户，`\h`代表主机名，`\w`代表工作目录，`\$`表示命令提示符号“\$”（\是转义符号，可以使用诸如\n、\t之类的）。对应屏幕上显示的命令行提示符为：dranyu@dranyu-desktop:~\$。

上述的`PS1`变量只显示了命令行提示符的一部分信息，shell 中还允许使用一些特殊字符（通常是“\”加各种字母表示）显示不同的信息，如命令的历史编号、日期和时间等。这些特殊字符如下所示。

- `!: 显示当前命令的历史编号。`
- `\#:` 显示此次 shell 中当前命令的编号。编号从 1 开始计算。
- `\$:` 显示命令提示符号，普通用户使用符号“\$”，而 root 用户使用符号“#”。此值的确定取决于当前用户的身份。
- `\W:` 显示当前工作目录的基名。例如，如果当前工作目录是`/usr/local/share`，那么这个值会显示为`share`。
- `\w:` 显示当前工作目录的完整路径。例如，如果当前工作目录是`/usr/local/share`，那么这个值会显示为`/usr/local/share`。
- `\[`: 后接一系列非打印字符。其可以用来添加一个终端控制序列到提示符中，例如更改颜色、添加闪烁效果或加粗。
- `\]:` 跟在非打印字符之后，与“`\[`”相呼应。
- `\`:` 显示反斜杠。
- `\d:` 显示星期、月份和日期的数字形式。如 8 月 23 日显示为“日 8 月 23”。
- `\h:` 显示运行此 shell 的计算机主机名。
- `\s:` 显示当前 shell 的名称。对于 Bash shell，这个值显示为 Bash。
- `\t:` 打印当前时间的时、分和秒。例如，20 时 9 分 19 秒显示为 20:09:19。
- `\u:` 显示当前使用 shell 的用户名。

只要将`~/bashrc`文件中的`PS1`变量修改了，就可以改变命令行提示符。将`PS1`变量修改



为如下所示：

```
PS1='[\u@\h:\w \t \! \$ ]'
```

保存文件之后，重新启动模拟终端，命令行提示符在模拟终端的显示屏幕上显示如下所示：

```
dranyu@dranyu-desktop:~ 20:21:56 501 $
```

字符串“20:21:56”表示当前时间为晚上 8 点 21 分 56 秒；数字“501”表示即将输入的命令的历史编号。

3. 设置命令别名

shell 提示别名的功能，通过别名使用几个字母就可以代替一条很复杂的完整的命令。

例 10.15：不带参数的 alias

使用不添加参数的 alias 命令可以查看当前 shell 使用了哪些别名，执行 alias 命令的执行结果如下所示：

```
dranyu@dranyu-desktop:~$ alias //输入 alias 命令
alias ls='ls --color=auto'
```

例 10.16：定义命令的别名

在终端可以使用 alias 定义命令的别名，将查看文件夹内容的命令 ls -l 命名为 ll，执行命令如下：

```
dranyu@dranyu-desktop:~$ alias ll='ls -l'
```

在此次 shell 中执行命令 ll 实际上等于执行命令 ls -l。

上述的例子中定义的别名仅限于在此次 shell 中有效，即此 shell 退出之后，再重新登录 shell 别名 ll 将不可使用。若希望自己的别名命令可以在每次登录 shell 的时候有效，用户必须在 ~/.bashrc 文件里添加 alias ll='ls -l' 记录行。若希望 Ubuntu 系统上的所有人都使用同样的自定义的别名，可以在配置文件 /etc/bashrc 上添加 alias ll='ls -l' 记录行。

此处简述在 ~/.bashrc 文件中定义别名，使用文本编辑器 Gedit 打开 ~/.bashrc 文件，必可找到一段代码如下所示：

```
# some more ls aliases
#alias ll='ls -l'
#alias la='ls -A'
#alias l='ls -CF'
```

将 alias 语句前的注释符号“#”去掉，即可激活该命令。如去掉 alias ll='ls -l' 行前的“#”符号，然后保存 ~/.bashrc 文件，重新启动终端。在命令行提示符上执行命令 ll 和命令 ls -l 的结果是一样的。

用户可添加新行到上述代码的后面如下所示：

```
alias workspace='cd ~/document/workspace'
alias rm='rm -i'
```

然后保存 ~/.bashrc 文件，重新启动终端。执行 workspace 命令，表示进入到 ~/document/workspace 目录，其执行结果与执行 cd ~/document/workspace 的结果是一样的。而第二行表示使用 rm 命令删除文件时，shell 会询问用户是否删除此文件，这有效地防止了用户错误地删除文件。

4. 添加环境变量

每个活动的 shell 都要存储一些它需要的信息，这些信息叫做环境变量。环境变量能够保存类似配置文件位置、邮箱和搜索路径目录的信息。它们还可保存 shell 提示符、历史列表大

小和操作系统类型。在终端上使用 set 命令可查看详细的环境变量信息。PATH 环境变量是在 Linux 的使用过程中使用率最高的，它是用来设定执行命令时系统的搜索路径的。

在终端上使用“echo \$(变量名)”命令可查看环境变量的值。执行 echo \$PATH 命令，执行结果如下所示：

```
dranyu@dranyu-desktop:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
```

若我们希望在执行命令时，系统会在~/bin 目录下搜索命令，我们可以在~/bashrc 文件里的结尾添加一段代码如下所示：

```
PATH=$PATH: $HOME/bin
```

其中\$PATH 是原来系统已配置的搜索路径，符号“:”是分隔符，\$HOME/bin 就表示当前用户主目录下的 bin 文件夹，即~/bin。变量 HOME 保存的是当前用户的主目录。

还可设置用户环境变量，设置一个变量 W 的值为~/document/workspace，同样在~/bashrc 文件的结尾添加一行代码如下所示：

```
W='~/document/workspace'
```

那么用户可通过 echo \$W 命令查看变量的值；而使用 cd \$W 命令则可进入到 ~/document/workspace 目录，执行结果与执行 cd ~/document/wrokspace 命令的结果一样。

10.4.4 编辑命令行

用户在输入命令行过程中总会出现错误，那么修改命令行就变得很重要。Bash shell 提供了方便的编辑命令行功能。

使用键盘上方向键←和→，可移动命令行的光标；使用键盘上方向键↑和↓，可获取历史命令列表中的命令。

使用组合键可快速移动光标，移动光标的常用组合键如表 10-1 所示。

表 10-1 移动光标

组合键	含义
Ctrl+A	将光标移动到命令行的开始位置
Ctrl+E	将光标移动到命令行的末尾位置
Ctrl+F	将光标向右移动一个字符
Ctrl+B	将光标向左移动一个字符
Ctrl+I	清屏并从屏幕的最上面开始新的一行

编辑命令行的常用组合键如表 10-2 所示。

表 10-2 编辑的组合键

组合键	含义
Ctrl+D	删除当前光标位置的字符
Ctrl+T	交换光标与光标前字符的位置
Ctrl+K	剪切光标到行末的字符串
Ctrl+U	剪切光标到行首的字符串



续表

组合键	含义
Ctrl+W	剪切光标前的一个单词
Ctrl+Y	粘贴最近剪切的字符串
Ctrl+C	删除整行，并换到下一行

10.4.5 命令行补全

Bash shell 提供了命令行自动补全功能，可对不完整的值进行补全，减少用户的输入。以下的值可以部分输入，然后按下 Tab 键，就可补全。

- 环境变量：当输入的字符串是以\$开头的，那么 shell 会在环境变量中搜索匹配的变量。
- 用户名：当输入的字符串是以~开头的，那么 shell 就以用户名补全文本。例如，系统用户 root、~r 字符可能补全为~root。
- 命令、别名或函数：当输入的字符串是以常规字符开头的，那么 shell 会尝试使用命令、别名或函数来补全字符串。
- 文件名：命令提示符上，当已输入命令，输入参数时只要输入的字符串是以/或常规字符开头的，那么 shell 会将字符串补全为目录或文件名。

这里举几个自动补全的例子，符号“//”后为解释说明。

例 10.17：补全环境变量

```
dranyu@dranyu-desktop:~$ echo $P      //此处再次按下 Tab 键
$PATH      $PPID      $PS2      $PWD
$PIPESTATUS $PS1      $PS4
```

例 10.18：补全目录或文件名

输入 Gedit /etc/profile 命令的一种补全过程如下所示：

```
dranyu@dranyu-desktop:~$ Gedit /e
按一次 Tab 键后，扩展为：
dranyu@dranyu-desktop:~$ Gedit /etc/
输入字符 pr，然后按下 Tab 键，发出“嘟”的声音，再按一次 Tab 键，屏幕显示结果为：
dranyu@dranyu-desktop:~$ Gedit /etc/pr
printcap  profile  profile.d/ protocols
然后可参考命令提示符下的文件名输入 profile 文件名
```

补全/etc 的时候，由于/目录只有 etc 子目录是以 e 字母开头的；而补全 profile 文件名时，目录/etc 下有多个以 pr 开头的文件或文件夹。

10.4.6 后台执行

交互式的执行命令是指一个命令执行完毕之后才能执行下一个命令。在 shell 中，使用& 符号将命令放在后台执行，例如在终端命令提示符后输入 apt-get install vim & 命令表示在后台执行安装 vim 命令，命令在后台运行，同时 shell 转为等待用户输入状态。

按下“Ctrl+Z”组合键停止一个已运行在前台的应用程序，在命令提示符后再输入 bg 命令，可将命令激活并放置到后台。而按下“fg”键就可以将后台的命令放置在前台来。通过 bg 和 fg 的命令切换前台和后台的应用程序。

例 10.19：命令前后台执行的切换过程

以下通过执行 Gedit ~/.bashrc & 命令详细解说后台执行与前台切换的执行过程：

(1) 在命令提示符后输入命令 Gedit ~/.bashrc &, 并按下回车键, 执行结果如下所示:

```
dranyu@dranyu-desktop:~$ Gedit ~/.bashrc & //输入命令
[1] 19084 //显示结果, 19084 是 Gedit 命令的进程号
dranyu@dranyu-desktop:~$
```

(2) 此时 Gedit 应用程序已打开了 ~/.bashrc 文件, 并且模拟终端仍然处于等待命令输入状态。在第三行的命令提示符后输入 fg 命令, 执行结果如下所示:

```
dranyu@dranyu-desktop:~$ fg //输入 fg 命令
Gedit ~/.bashrc
```

(3) 此时模拟终端的命令提示符没有了, 屏幕上显示了处于前台的命令 Gedit ~/.bashrc。为了将命令 Gedit ~/.bashrc 置于后台执行, 在终端上按下“Ctrl+Z”组合键, 终端屏幕上显示执行结果如下所示:

```
^Z //按下 Ctrl+Z 组合键
[1]+ Stopped Gedit ~/.bashrc
dranyu@dranyu-desktop:~$
```

(4) Stopped 字符串表示命令 Gedit ~/.bashrc 进入了停止状态, Gedit ~/.bashrc 不可编辑, 但命令已切换至前台, shell 处于等待命令输入的状态。在命令提示符后输入 bg 命令, 按下回车键, 屏幕上显示执行结果如下所示:

```
dranyu@dranyu-desktop:~$ bg
[1]+ Gedit ~/bashrc &
dranyu@dranyu-desktop:~$
```

(5) 命令 bg 激活了 Gedit ~/bashrc 命令, 此时 Gedit 可编辑~/bashrc 文件。整个操作过程如图 10-3 所示。

```
dranyu@dranyu-desktop: ~
文件(E) 编辑(E) 查看(V) 终端(T) 帮助(H)
dranyu@dranyu-desktop:~$ gedit ~./bashrc &
[1] 19849
dranyu@dranyu-desktop:~$ fg
gedit ~./bashrc
^Z
[1]+ Stopped gedit ~./bashrc
dranyu@dranyu-desktop:~$ bg
[1]+ gedit ~./bashrc &
dranyu@dranyu-desktop:~$
```

图 10-3 命令前后台切换

当命令处于后台执行时, 用户可使用命令 jobs 查看后台命令的运行情况, 执行命令如下所示:

```
dranyu@dranyu-desktop:~$ jobs
[1]- Running Gedit ~/bashrc &
[2]+ Stopped vim
```

其中 Running 和 Stopped 表示命令的运行状态, [1]和[2]表示命令在后台的任务编号。对于后台有多个任务在运行的情况, shell 提供了符号“%”访问指定的任务。若要将命令 vim 放置于前台运行, 可执行 fg %2 命令或 fg %vim 命令。



补充说明

符号%的用法如下所示：

- (1) %：单独使用%表示最近一个被放置到后台的任务。执行%命令与执行 fg 命令的效果是一样的，与 fg %命令也是一致的。
- (2) %string：代表任务列表中以特定 string 字符串开头的任务。string 必须是任务中唯一的。
- (3) %number：数字 number 表示任务的序列号。
- (4) %?string：代表任务列表中包含 string 字符串的任务。
- (5) %--：代表在最近停止任务之前停止的任务。使用 jobs 命令时，任务编号后的符号“-”指定为在最近停止任务之前停止的任务；而符号“+”指定为最近停止的任务。

10.4.7 退出 shell

退出 shell 最常用的方法是在提示符后输入 exit 命令。按下 Ctrl+D 组合键，同样可以退出 shell。

当退出 shell 的时候，即模拟终端关闭时，Bash shell 会执行一个配置文件，此文件为 ~/.bash_logout。此文件通常只包含一个操作，那就是将屏幕清除。如果想在退出 shell 前执行一些工作，例如清空缓冲区或执行备份，都可以在此文件中设置。

使用 Gedit ~/.bash_logout 命令，查看文件内容如下所示：

```
# ~/.bash_logout: executed by bash(1) when login shell exits.  
# when leaving the console clear the screen to increase privacy  
if [ "$SSHLEVEL" = 1 ]; then  
    [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q  
fi
```

然而那些在登录 shell 中通过输入 shell 命令（如 tsh, ksh）启动的 shell，在退出时不会执行 ~/.bash_logout 文件。它们退出后，控制权转交创建它们的 shell。

10.5 常用的 Linux 命令



上述我们讲述了如何启动终端，以及如何编辑命令行的相关知识，此节将讲述 Ubuntu 中常用的命令。这些常用的命令包括目录和文件的命令、帮助命令、文档的打包和压缩命令，以及用户的系统操作命令。

10.5.1 基本操作命令

在此之前，我们都使用了一些 Ubuntu 的基本操作命令，如浏览目录内容的命令 ls 和进入到指定目录的命令 cd。接下来结合例子说明基本操作命令的用法。

1. 浏览目录内容命令 ls

命令 ls 能够罗列出目录的所有内容，其常用选项如下所示。

- -a：显示指定目录下的所有文件及目录，包括隐藏文件和目录（以点“.”开头的文

件和目录是隐藏文档), 还包括本级目录“.”和上一级目录“..”。

- **-l:** 将文件和目录详细列出来, 包括文件状态、权限、拥有者、文件大小和文件名等。
- **-R:** 递归显示下层子目录。
- **-help:** 显示命令 ls 的帮助信息。

命令 ls 的操作目录在默认的情况下是当前的工作目录, 也可通过命令 ls 的参数来指定。

例 10.20: 以详细列表的形式罗列出主目录下的文件和目录

使用 ls 命令罗列主目录下的文件, 执行命令如下:

```
/ $ls -l ~或  
~$ls -l
```

2. 改变工作目录命令 cd

命令 cd 能够改变工作目录, ppxx 的语法如下所示:

```
cd [路径]
```

使用命令 cd, 要知道何为绝对路径, 以及何为相对路径。若路径参数是以“/”开头的, 那么此路径为绝对路径, px 是从根目录开始的, 如/usr/share。若路径参数不是以“/”开头的, 则此路径为相对路径。

在 Ubuntu 中, 路径参数有几个比较特殊的目录:“.”、“..”、“~”、“~用户名”和“-”。 “.” 表示当前路径; “..” 表示当前工作目录的上一级目录; “~” 表示当前使用 shell 用户的主文件夹; “~用户名” 表示指定用户名的主文件夹; “-” 表示返回到上次执行命令 cd 之前的目录。这几个特殊目录使用的具体操作如图 10-4 所示。

```
dranyu@dranyu-desktop:~/homes$ cd ..  
dranyu@dranyu-desktop:/$ cd  
dranyu@dranyu-desktop:~$ cd /usr/bin/  
dranyu@dranyu-desktop:/usr/bin$ cd .  
dranyu@dranyu-desktop:/usr/bin$ cd ~  
dranyu@dranyu-desktop:~$ cd .  
/usr/bin  
dranyu@dranyu-desktop:/usr/bin$ cd ~d  
-daemon/ ~dovecot/ ~dranyu/  
dranyu@dranyu-desktop:/usr/bin$ cd ~dranyu/  
dranyu@dranyu-desktop:~$
```

图 10-4 命令 cd 的操作

图中的第二行是不带参数的命令 cd, 它表示返回到当前用户的主文件夹, 与命令“cd ~”的效果是一样的。这种设置可让用户快速返回到自己的主文件夹。Ubuntu 中用户的多数文档都是保存在用户的主文件夹中。

3. 显示文本文件内容命令 cat

命令 cat 可用来显示文本文件的内容。命令 cat 只是将文本文件的内容显示在屏幕上, 或通过重定向将文本文件的内容读取出来并写入其他文件。其常用选项如下所示。

- **-n:** 在所有的文本行前显示行号, 以 1 开始。
- **-d:** 只计算所有的非空行, 将行号显示在行首, 以 1 开始。
- **-e:** 在每行的行尾显示\$符号。

例 10.21: 使用不带参数的 cat 命令显示 hello1 文件的内容

```
dranyu@dranyu-desktop:~/test$ cat hello1  
hello  
hello1
```



例 10.22：显示文件内容的同时显示行号和行尾的\$符号

```
dranyu@dranyu-desktop:~/test$ cat -en hello1
1 hello$
2 hello1$
```

命令 `cat` 还可以同时显示多个文件，用法形式如：`cat file1 file2 file3`，命令 `cat` 的选项在此同样有效。

例 10.23：同时显示多个文件

假设文件 `hello1` 的内容包括两行：“hello”和“hello1”，文件 `hello2` 只有一行内容为 `hello 2`，使用命令 `cat` 同时显示 `hello1` 和 `hello2`，执行结果如下所示：

```
dranyu@dranyu-desktop:~/test$ cat -en hello1 hello2
1 hello$
2 hello1$
3 hello2$
```

由此可知，`-n` 和 `-b` 是对所有需要处理的文件进行计算的。

4. 匹配字符串命令 grep

命令 `grep` 可以在文本内容中搜索与所给匹配模式相匹配的文本行，并将文本行显示在屏幕上，其语法如下所示：

```
grep [选项] 匹配模式 [文件列表]
```

例 10.24：在文件 `hello1` 中搜索包含字符“1”的所有文本行

```
dranyu@dranyu-desktop:~/test$ grep 1 hello1
hello1
dranyu@dranyu-desktop:~/test$ cat hello1 | grep 1
hello1
```

上述两条命令获得了同样的结果，都是从文件 `hello1` 中搜索含有字符“1”的文本行，并显示在屏幕上。第一条命令将文件 `hello1` 作为操作参数，命令 `grep` 直接在 `hello1` 搜索与字符“1”匹配的文本行；而第二条命令 `grep` 的操作文本来自标准输入，即命令 `cat hello1` 的输出。

5. 排序命令 sort

命令 `sort` 将文本中的各行进行排序，其语法如下所示：

```
sort [选项] [文件列表]
```

命令 `sort` 的常用的选项如下。

- `-d`: 按字典顺序排序，比较时仅对字母、数字、空格和制表符有意义。
- `-f`: 忽略大小写。
- `-r`: 按逆序排序输出排序结果。

命令 `sort` 将逐行对文件中的内容进行排序，如果两行的首字符相同，该命令将继续比较这两行的下一字符，如果还相同，将继续进行比较，直到排好序。

例 10.25：对例 10.22 中的 `hello1` 文件进行逆序排序

```
dranyu@dranyu-desktop:~/test$ sort -r hello1 //使用-r 选项
hello1
hello
或
dranyu@dranyu-desktop:~/test$ cat hello1 | sort -r //使用管道“|”命令
hello1
hello
```

6. 分页命令 less 或 more

屏幕显示信息的行数是有限的，而且其为命令行的标准输出。为了能够方便浏览命令的输出信息，Linux 系统提供了分页的命令。命令 less 和 more 就担任着分页的角色。这两个命令在显示文本的时候会将信息显示一屏之后就停顿下来，然后接收到显示下一屏的命令时，将显示文本的下一屏信息。

虽然这两个命令很相似，但还是存在一些细微的差别。命令 less 执行之后，到文件的结尾显示 EOF 字样，按下 Q 键才会返回到 shell；而命令 more，执行到文件结尾就返回到 shell。命令 less 的分页功能可接受方向键、字母 J 和 K 作为屏幕滚动的命令；而 more 接受的滚动命令是空白键、字母 S 和 D。在命令 less 的执行期间，还提供了搜索功能。命令 less 中使用搜索功能的方法为：按下符号 “/”，然后输入指定的字符串，最后按下回车键，此时屏幕上就会显示与输入的字符串相匹配的文本行。

例 10.26：使用 less 或 more 查看文件 hello1

```
dranyu@dranyu-desktop:~/test$ less hello1
或 dranyu@dranyu-desktop:~/test$ more hello1
```

命令 less 或 more 除了将文件作为处理的内容外，还可将标准输入作为处理的内容。

例 10.27：将命令 ls 的标准输出通过管道输入到命令 less 或 more 中

```
dranyu@dranyu-desktop:~/test$ls | less
或 dranyu@dranyu-desktop:~/test$ls | more
```

10.5.2 帮助命令

Ubuntu 系统中保存了大量的命令使用帮助信息，因此当用户对系统中的命令不是很熟悉或不会使用时可查看相关的使用帮助信息。这一小节将介绍几个常用的帮助命令，包括：man 查询其他命令的使用方法、help 查看系统中 shell 的命令、使用 whereis 查看命令的位置，以及命令其他使用信息、info 命令的使用方法。

1. 显示命令的帮助信息命令 man

对于每个 Linux 系统来说，命令 man 都为 Linux 命令提供了联机文档。命令 man 的名称是 manual 的缩写。使用 man 显示的命令使用手册都包括命令的语法、各选项的意义以及相关命令的等。命令 man 的语法结构如下所示：

```
man [选项] 命令名称
```

命令 man 的常用选项如下所示。

- **-s：**根据章节显示。一个命令名称可能会有很多类，使用此选项可指定显示的章节。
例如，显示命令 mv 第一章节的内容，使用命令为：man -s 1 mv。
- **-w：**不显示手册，只显示文件所在的位置。
- **-a：**显示所有手册页，而不是只显示第一个。

命令 man 是以章节来显示命令的帮助信息的，而且在 man 显示的手册中第一行会显示命令章节的信息标签数字，这些数字可作为命令 man 的参数指定显示的章节。章节的具体含义如表 10-3 所示。



表 10-3 手册中的章节信息

章节数字	说 明
1	一般用户的命令
2	系统调用的命令
3	C 语言函数库的命令
4	有关驱动程序和系统设备的解释
5	配置文件的解释
6	游戏程序的命令
7	其他软件或程序的命令和系统维护命令

例 10.28：查看 cp 命令在 man 使用手册中第一章

```
dranyu@dranyu-desktop:/$man -s 1 cp
```

-s 和 1 都是参数，cp 为所要查看的命令。

例 10.29：查看命令 cp 说明文档的位置

```
dranyu@dranyu-desktop:/$ man -w cp  
/usr/share/man/man1/cp.1.gz
```

2. 系统帮助命令 help

命令 help 可用于查看所有系统 shell 命令。此命令使用格式如下所示：

```
help [命令]
```

其中的“命令”选项表示 shell 命令的名称。

例 10.30：使用 help 显示系统的 shell 命令

```
dranyu@dranyu-desktop:~$ help  
GNU bash, version 3.2.48(1)-release (i486-pc-linux-gnu)  
shell 的命令是内置定义的，输入 “help” 可以看到这个列表  
输入 “help name” 来查找关于 “name” 命令功能的更多信息  
输入 “info bash” 来查看关于 shell 的更多信息  
输入 “man -k” 或者 “info” 来查看没有列在这个列表中的命令的信息  
带有星号(*)的命令是被禁止的
```

```
JOB_SPEC [&]                                (( expression ))  
. filename [arguments]                      :  
[ arg... ]                                     [[ expression ]]  
alias [-p] [name[=value] ... ]      bg [job_spec ...]  
....  
....  
while COMMANDS; do COMMANDS; done { COMMANDS ; }
```

例 10.31：获取列表中命令 alias 的更多信息

```
dranyu@dranyu-desktop:~$ help alias  
alias: alias [-p] [name[=value] ... ]  
`alias' with no arguments or with the -p option prints the list  
of aliases in the form alias NAME=VALUE on standard output  
otherwise, an alias is defined for each NAME whose VALUE is given  
A trailing space in VALUE causes the next word to be checked for  
alias substitution when the alias is expanded. Alias returns  
true unless a NAME is given for which no alias has been defined
```

3. 查看命令所在位置信息的命令 whereis

命令 whereis 可寻找指定命令所在的位置。命令 whereis 一般不使用选项参数，但是必须指定命令名称。

例 10.32：查看 cp 命令相关信息的位置

```
dranyu@dranyu-desktop:/$ whereis cp
cp: /bin/cp /usr/share/man/man1/cp.1.gz
```

其中字符串“/bin/cp”表示命令 cp 的可执行文件保存在/bin 目录下，而字符串“/usr/share/man/man1/cp.1.gz”

例 10.33：显示 whereis 的帮助信息

```
dranyu@dranyu-desktop:~$ whereis --help
whereis [ -sbmu ] [ -SBM dir ... -f ] name...
```

4. 查看 GNU 程序命令的帮助文档命令 info

Ubuntu 的许多命令都来源于 GNU，因此 GNU info 命令通常要比 man 提供了更完整的命令使用信息。

命令 info 不带任何参数时，在屏幕上显示系统的所有手册页的分类索引。

当命令 info 后接一个指定的命令，那么屏幕会显示指定命令的信息手册页。按下 Q 键，可退出手册，返回到 shell。按下“Pgup(上页)”或“PgDn (下页)”键，就可向上或向下翻屏幕的信息页。按下 S 键，可在手册中查找指定的字符串。

例 10.34：查看命令 grep 的使用方法

```
dranyu@dranyu-desktop:~$ info grep
```

执行命令后，在终端的屏幕上显示 grep 的 info 帮助手册页，如图 10-5 所示。

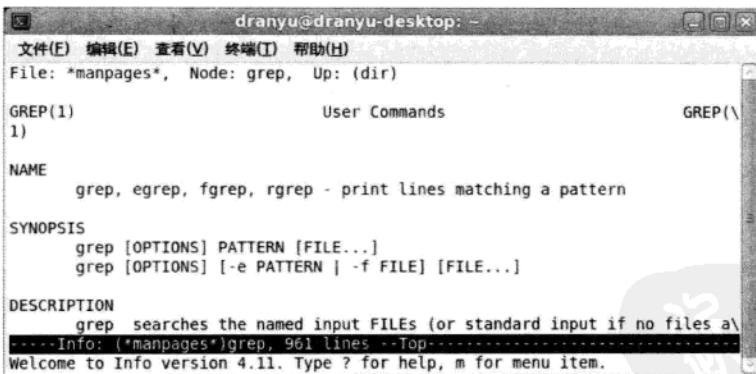


图 10-5 显示 grep 的 info 帮助手册页

补充说明

一般在程序员设计命令的时候，大部分的命令都会拥有一个调用命令使用说明的选项，那就是--help 选项或-h 选项。当用户使用一个命令的时候，可使用这一选项，其使用方法为：command --help 或 command -h。例如查看 cp 的使用方法，可使用命令：cp --help。



10.5.3 文件和目录操作命令

Ubuntu 中提供了多种操作文件的命令，例如，创建文件的命令、复制文件的命令 cp、移动文件的命令 mv 和删除文件的命令 rm 等。

1. 创建文件的命令

在 Linux 下，有许多的命令可创建文件，如 touch、echo 和 cat 等。

例 10.35：在工作目录下创建一个空文件 newfile

使用命令 touch、echo 和 cat 创建新文件，执行如下命令：

```
touch newfile
```

或

```
echo > newfile
```

或

```
cat > new file
```

后两条命令使用了重定向标准输出。

2. 复制文件和目录命令 cp

在配置新的文件之前，有必要对文件进行备份，则可使用复制文件的命令 cp。若需要复制一个文件到指定的目录，可使用命令格式如下所示：

```
cp 源文件 目标路径
```

例 10.36：将文件/etc/profile 复制到用户的主目录

```
dranyu@dranyu-desktop:/$cp /etc/profile ~
```

此种格式的源文件可以是多个文件或是可扩展的文件名。

例 10.37：将/etc 目录下以 Bash 开头的文件复制到主目录的 test 目录下

```
dranyu@dranyu-desktop:/$cp /etc/bash* ~/test/
```

命令 cp 还可以将文件在复制的过程中重命名，其命令格式如下所示：

```
cp 源文件 目标文件
```

例 10.38：备份配置文件 passwd

在修改一个配置文件 passwd 之前，先将文件 passwd 进行备份，将备份文件命名为 passwd.bak，执行如下命令：

```
dranyu@dranyu-desktop:/$cp passwd passwd.bak
```

使用命令 cp -help 可查看更多的使用细节。

3. 删除文件和目录命令 rm

命令 rm 可从文件目录中删除文件及整个目录。在默认情况下，命令 rm 只能删除指定的文件，而不能删除目录。例如删除当前工作目录的所有以.png 为后缀的文件，可执行命令如下：

```
rm *.png
```

若要删除指定的目录，就必须使用-r 选项。

例 10.39：删除用户主目录下的 test 目录

```
dranyu@dranyu-desktop:/$rm -r ~/test
```

使用命令 rm 删除的文件是不可被恢复的，用户在使用时必须小心。命令 rm 提供了交互式删除模式，使用-i 选项完成此任务。即在执行删除文件前，信息提示用户是否真的需要删除文件，键入 y 开头的字符串，则表示肯定执行删除操作；键入其他字符串则表示不执行操作。

例 10.40：交互式删除图片文件 pice.jpg

```
dranyu@dranyu-desktop:~/test$ rm -i pice.jpg
```

```
rm: 是否删除普通空文件 "pice.jpg" ? y
```

输入“y”字符表示删除 pice.jpg 文件。

使用命令 rm -help 可查看更多的使用细节。

4. 移动文件和目录命令 mv

命令 mv 可将一个文件从一个目录中移动到指定的目录中。若在目标目录下不存在此文件，文件会被创建；若在目标目录下存在同名的文件，那么已存在的目标文件会被覆盖。

使用-i 选项，在出现文件覆盖时，系统询问是否重写，要求用户回答 y 或 n，这样可以避免误覆盖文件。

例 10.41：使用命令 mv 对文件进行重命名

在当前工作目录下将 hello 2 重命名为 hello1，执行命令如下所示：

```
dranyu@dranyu-desktop:~/test$ mv -i hello2 hello1
mv: 是否覆盖 "hello1" ?y
```

此时 hello1 文件已存在，输入 y，然后按回车键，执行命令，覆盖 hello1 文件。与选项-i 相反的选项是-f，这一选项表示强行禁止交互操作。在 mv 操作要覆盖已有的目标文件时系统不会给出任何提示，并且执行命令 mv 的操作，也就是说指定此选项后，i 选项将不再起作用。

5. 创建目录命令 mkdir

目录就是为用户更好地管理文件而出现的。因而在管理 Linux 的时候用户必须掌握创建目录的命令 mkdir，其的命令格式如下所示：

```
mkdir [参数选项] 目录名称
```

例 10.42：使用-m 选项可设置新建目录的访问权限

在当前的工作目录下创建一个只允许目录的拥有者读、写和执行的目录 test，其命令执行如下所示：

```
dranyu@dranyu-desktop:~$ mkdir -m 700 test
dranyu@dranyu-desktop:~$ ls -ld test
drwx----- 2 dranyu dranyu 4096 2009-08-27 16:44 test
```

第二行的命令是为了显示命令 mkdir 的执行结果。

补充说明

关于文件和目录的权限问题在后续章节中会讲述。上例的第三行的“drwx-----”字符串后面 9 个字符表示的就是此目录 test 的访问权限。此 9 个字符分为 3 组：前 3 个字符表示文件或目录拥有者的权限；中间 3 个字符表示与拥有者同组的其他用户的权限；最后的 3 个字符表示系统中其他组中用户的权限。3 组中都有 3 个字符，r 表示读权限、w 表示写权限和 x 表示执行权限。访问权限都可用数字来表示。如 r- 的数值是 4，-w- 的数值是 2，--x 的数值是 1。三者组合的时候就将相应的数值相加即可，如 rw- 的数值是 6，数字 6 是由 4 加 2 得来的。

10.5.4 文档打包和压缩命令

对磁盘上的文件进行打包和压缩，可方便用户管理文档，以及节省磁盘空间。将多个文件组合成一个文档，就需使用到 Linux 下的打包命令 tar。Ubuntu 下提供了多种压缩工具，如 gzip。



1. 文档打包命令 tar

命令 tar 可对文件和目录进行打包。使用命令 tar 可将多个文件打包到一个文档中，也可将一个文件添加到一个已存在的包文档中。命令 tar 的命令格式如下所示：

```
tar [主选项+辅选项] 文件或目录
```

在使用命令 tar 时，主选项是必需的，它告诉 tar 要做什么事情，它的主要选项参数如表 10-4 所示；辅选项则是可选的，常用的参数如表 10-5 所示。

表 10-4 命令 tar 主选项的常用选项

选 项	说 明
-c	创建新的包
-x	从文档中释放文件
-r	将文件加入已有的包
-t	列出包中所有文件
-u	更新包中的文件

表 10-5 命令 tar 辅选项的常用选项

选 项	说 明
f	指定包的文件名
v	显示执行过程
z	指明待操作的包是一个 gzip 包

例 10.43：将文件打包为 rar 文档

将~/picture 目录中所有以“jpg”为后缀的文件打包到 picture.tar 文档中，将 picture.tar 保存在当前工作目录，可执行命令：

```
tar -cvf picture.tar ~/picture/*.jpg
```

例 10.44：显示 rar 文档中文件列表信息

若要显示文档 picture.tar 中的内容，则将上述命令中的 c 修改为 t 即可，执行命令为：

```
tar -tvf picture.tar
```

例 10.45：将 rar 文档解压到当前的工作目录

若要释放已存在的文档，可将上面命令选项中的 t 修改为 x，这样将包文档中的文件释放到当前的工作目录。若要将文件释放到指定的目录~/bak，可使用命令：

```
dranyu@dranyu-desktop:~$tar -xvf picture.tar ~/bak
```

命令 tar 还可以在对多个文件打包的同时进行 gzip 压缩，则上述创建文档的命令如下所示：

```
dranyu@dranyu-desktop:~$tar -czvf picture.tar.gz ~/picture/*.jpg
```

对此包所需进行的信息查看和释放操作相应地添加选项 z 即可。

2. 压缩命令 gzip

命令 gzip 是在 Linux 中经常使用的一个对文件进行压缩和解压缩的命令，它的命令格式为：

```
gzip [选项] 压缩(解压缩)的文件名
```

命令 gzip 压缩文件，生成一个以.gz 为后缀的压缩文档。

例 10.46：压缩单个文件

若需要压缩~/document/doc.txt 文件，可使用如下命令：

```
dranyu@dranyu-desktop:~$ gzip ~/document/doc.txt
```

这一操作使得在当前工作目录生成了一个名为 doc.txt.gz 的文档。

例 10.47：解压缩文档

若要对此文件进行解压缩，可使用如下命令：

```
dranyu@dranyu-desktop:~$ gzip -d doc.txt.gz
```

执行命令之后，在当前工作目录上将出现 doc.txt 文件，而 doc.txt.gz 就被删除了。

10.5.5 系统操作命令

Ubuntu 中提供了许多图形界面方便用户进行系统操作，其实，在执行命令时 Ubuntu 图形界面程序调用了 shell 的系统操作命令，这些系统操作命令包括：注销用户、切换用户、关闭系统和重启系统。

1. 注销用户命令 logout

在使用注销用户命令的时候，需要说明一下用户登录命令 login。使用命令 login 可在 shell 中登录系统，在 shell 中登录系统之后，此 shell 就变为了登录 shell。而命令 logout 是命令 login 的反操作，其用来注销使用命令 login 登录的用户。

例 10.48：在终端上登录系统并退出

在终端使用命令 login，以 sunny 的身份登录系统，然后再使用命令 logout 注销 sunny 用户，如图 10-6 所示。

The screenshot shows a terminal window titled 'root@dranyu-desktop: /home/dranyu'. The session starts with the user dranyu running a sudo su command to become root. The root shell then runs the login command for the user sunny. The terminal displays the standard Ubuntu login message, which includes a copyright notice and a link to official documentation. Finally, the user logs out using the logout command, returning to the root shell.

```
root@dranyu-desktop:~$ sudo su
[sudo] password for dranyu:
root@dranyu-desktop:/home/dranyu# login
dranyu@dranyu-desktop:~$ dranyu@dranyu:~$ login
dranyu@dranyu:~$ 用户名 : sunny
口令 :
上一次登录: 五 08月 28 20:59:21 CST 2009 pts/0 上
Linux dranyu@dranyu 2.6.28-14-generic #47-Ubuntu SMP Sat Jul 09 16:06:06 UTC 2011
The programs included with the Ubuntu system are free software.
the exact distribution terms for each program are described
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent per
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/

24 packages can be updated.
48 updates are security updates.

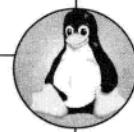
sunny@dranyu@dranyu:~$ logout
root@dranyu@dranyu:~$ logout
root@dranyu@dranyu:~$
```

图 10-6 命令 login 和 logout

2. 关闭系统的命令 shutdown 和 halt

命令 shutdown 和 halt 都可执行关闭系统的操作，它们的执行需要拥有系统管理员的权限。其命令 halt 调用 shutdown -h 命令。命令 shutdown 常用的功能举例如下所示。

- shutdown -h now：现在立即关机。
- shutdown -r now：现在立即重启计算机。
- shutdown -r +5：系统将在五分钟之后重启。



- shutdown -h +3：“系统将在 3 分钟后关闭”——提示使用者计算机将在 3 分钟后关闭。命令 halt 执行时，会先关闭应用进程，然后执行 sync 系统调用，等待文件系统写操作完成之后停止内核。

3. 重启系统命令 reboot

在终端中，用户可执行命令 reboot 重启系统，用户必须拥有 root 的权限。命令 reboot 执行后，立刻关闭应用程序，然后退出系统，最后重启系统。

10.6 小结



使用命令可提高工作效率，因此熟悉和掌握命令的使用方法是十分重要的。本章的开始部分讲述了 shell 中命令行的语法及其编辑方法，可帮助读者了解 shell 解释器和掌握命令行的使用方法。本章的后半部分通过实例讲解 Linux 系统中常用的命令，以此让读者更深入地了解 Linux 命令的使用。

对于不熟悉的命令，读者可使用帮助命令。帮助命令 man 提供了比较完整的命令解释说明，命令的 man 手册包括了命令的语法、参数的详细介绍和一些常用的例子。本章介绍的基本操作命令、文件和目录操作命令、文档打包和压缩命令与系统操作命令都可在 man 手册中找到相关的使用说明。

10.7 习题



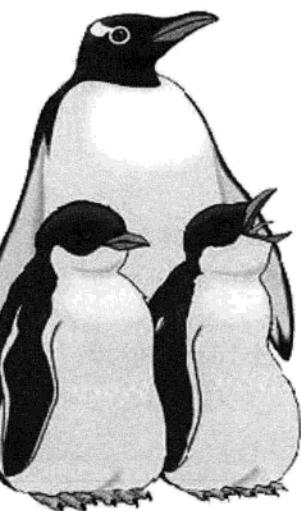
1. 启动模拟终端，使用命令 ls 罗列目录中的内容。
2. 使用命令 cd 进入到/etc 目录，使用命令 ls 列出与 Bash 匹配的文件（提示配合使用 shell 的通配符）。
3. 使用命令 cd 进入根目录，在此目录中查找包含字符 s 的文件或文件夹，对查询结果进行反向排序之后保存到主目录下的一个普通文件里（提示：使用管道，连接命令 ls、命令 grep 和命令 sort，同时使用标准输出重定向将结果重定向到指定的文件）。
4. 在主目录上，使用命令 mkdir 创建一个名为 doc 的目录，进入此目录，然后使用命令 touch 创建 3 个文件：test1.txt、test2.txt 和 test3.txt。使用命令 echo 向文件 test1.txt 添加文本“string1”，然后同样使用命令 echo 向文件 test1.txt 追加文本“string 2”。使用命令 cat 显示文件 test1.txt 的内容，检查文本是否添加到文件中。
5. 使用命令 tar 打包由上题创建的 doc 目录下的所有文件，然后使用命令 gzip 进行压缩。创建压缩包之后，使用 rm 删除以 t 开头的 3 个文件。命令 tar 对压缩包进行解压，释放出来的文件保存在当前工作目录。
6. 在主目录下使用命令 mkdir 创建一个 doc_bak 文件夹，然后进入到 doc_bak 目录，使用命令 cp 将第 4 题中 doc 目录下的文件复制到 doc_bak 文件夹。

第 11 章

软件包管理

软件包是一个在 Linux 下的常用术语。Ubuntu 中有很多软件，无论是社区开发提供的还是第三方提供的，都会被打包为软件包而发布。软件包的安装对每个用户来说，都是必须要执行的操作。开始的时候，Linux 软件是以通过开发者编译为二进制代码之后，压缩为文档包的形式而发布的，用户安装软件极不方便，而且对普通用户来说，还不一定能安装成功。这样软件包管理机制就成了 Linux 研究的一个课题。

本章主要讲述 Linux 下的软件包管理机制 Deb、rpm 和 tgz，通过几个实例讲述软件包的安装过程。软件包安装的方法可分为字符界面安装和图形界面安装。字符界面安装方式执行效率高，图形界面安装方式操作简单，而且图形界面安装方式在执行命令的时候，调用了字符界面安装方式的命令。





11.1 软件包的概述

软件包管理可以大大减少用户管理软件的麻烦操作。然而不同的发行版本对软件包的管理有着不同的管理机制，它们都要考虑两个问题，一是方便软件的安装和卸载，二是希望借此解决软件安装过程中经常遇到的软件包之间互为依赖的问题。目前，比较流行的有两种软件包：Deb 软件包——一种 Debian 系统，以及其衍生版本使用的管理机制；Rpm 软件包——一种广泛使用于 Red Hat、OpenSuse 等系统的软件包管理机制。一些其他的 Linux 发行版本也发布自己应用的软件包管理工具。例如，Slackware 是 Linux 历史上存活最久的发行版，而且一直保持着 KISS 的设计哲学，它的软件包不同于 Debian 的 Deb 和 Red Hat 的 rpm，而是通常的.tgz 格式文件再加上安装脚本。

11.1.1 DEB 软件包

Linux 发展初期，Linux 下软件包的安装工作唯有工程师可以胜任，用户群也是熟悉计算机的人群，普通用户是望而却步的。由于开源，每个软件的发布没有一个统一的标准，并且都是可以编译的源码包发布。这样安装软件就受到环境不同的影响，软件的安装很多时候是很难安装成功的。面对这样的情况，Debian Linux 首先提出“软件包”的管理机制——Deb 软件包，这种软件包管理机制将应用程序的二进制文件、配置文档、man/info 帮助页面等文件合并打包在一个文件中，然后用户使用软件包管理器直接操作软件包，在软件包管理器中完成获取、安装、卸载、查询等操作。

然而提出了软件包的概念，使用软件包来安装软件，并不能完全解决问题。随着 Linux 系统下软件数量呈级数增长，软件之间的依赖关系越来越复杂，其阻碍着 Linux 的发展，不能使 Linux 成为家喻户晓的操作系统。为了方便用户的使用，简化安装应用软件包，Debian Linux 开发出了 APT 软件包管理器。它能够自动检查和修复软件包之间的依赖关系。并且利用 Internet 网络带来的快捷的联通手段，APT 工具可以帮助用户主动获取软件包。因此，APT 工具再次促进了 Deb 软件包更为广泛地使用，成为 Debian Linux 的一个无法替代的亮点，并为以后衍生版本的 Linux 开创了一条光明的大道，为 Ubuntu 的流行奠定了基础。

Ubuntu Linux 继承了 Debian Linux 的 Deb 软件包的管理方法。Deb 软件包管理包括以下几个方面：软件包的类型，Deb 软件包的命名方式，软件包的优先级关系，软件包的状态，软件包的依赖性关系。

1. 软件包的类型

Deb 软件包包含了二进制可执行文件、库文件、配置文件和 man/info 帮助页面等文档。这样发布 Deb 软件包，用户才能查看相关的包信息，以及软件包之间的依赖关系。通常 Debian 包文件的后缀为.deb，因此称为 Deb 软件包。

Deb 软件包管理机制中，在 Ubuntu 下两种类型的软件包：二进制软件包（deb）和源码包（deb-src）。

(1) 二进制软件包 (Binary Packages): 二进制包以 .deb 扩展名来表示, 这些文件通常称为 DEB 文件, 其中包含可执行文件、文档, 配置文件和版权信息及其他一些东西。可以使用 Ubuntu 的 dpkg 工具解包安装。

(2) 源码包 (Source Packages): 包含软件源代码、版本修改说明、构建指令, 以及编译工具等。先由 tar 工具归档为.tar.gz 文件, 然后再打包成.dsc 文件。

用户不确定一个软件包具体类型时, 可以使用 file 命令查看文件类型。例如下面命令用于证实一个软件包的文件类型是否是 Deb 软件包文件。

```
*@*:~$ file mouseemu_0.16-0ubuntu3_i386.deb  
mouseemu_0.16-0ubuntu3_i386.deb: Debian binary package (format 2.0)
```

format 2.0 表示此软件包的打包方式使用了软件包格式的 2.0 版本。

同时, 我们也可以使用 ar 将软件包解压然后使用 ls 命令查看一下 Deb 软件包所包含的文件, 命令执行过程和执行结果如下所示:

```
*@*:~/Documents/dpkg$ ar -x mouseemu_0.16-0ubuntu3_i386.deb  
*@*:~/Documents/dpkg$ ls -l  
总用量 44  
-rw-r--r-- 1 dranyu dranyu 2418 2009-06-18 01:26 control.tar.gz  
-rw-r--r-- 1 dranyu dranyu 15096 2009-06-18 01:26 data.tar.gz  
-rw-r--r-- 1 dranyu dranyu 4 2009-06-18 01:26 debian-binary  
-rw-r--r-- 1 dranyu dranyu 17706 2009-01-13 06:04 mouseemu_0.16-0ubuntu3_i386.deb
```

可以发现其中几个文件: debian-binary, control.tar.gz、data.tar.gz。其中 debian-binary 是 Deb 软件包的神奇所在, 它包含所使用软件包格式的版本号。control.tar.gz 文件是包含了软件包管理工具所需信息的 tarball 文件。data.tar.gz 包括这个 Debian 软件包的覆盖区, 即 root 文件系统中的位置。用 dpkg 解包软件包, 而不配置就等同于将这个 tarball 文件解压到/目录。

在 Ubuntu Linux 中, 需要说明一个概念——虚拟软件包。将系统中具有相同或相近功能的多个软件包作为一个软件包集合, 称为虚拟软件包, 并指定其中一个软件包作为虚拟软件包的默认首选项。提出虚拟软件的意图就是为了防止软件安装过程中发生冲突。

2. 软件包的命名

在 Ubuntu Linux 中, 软件包的命名遵循以下约定:

```
Filename_Version-Reversion_Architecture.deb
```

这一约定集成了 Debian 的命名约定:

```
<foo>_<版本号>-<Debian 修订号>.deb
```

其中, Filename 表示软件包文件名, Version 表示软件版本号, Reversion 表示 Ubuntu 的修订版本号, Architecture 表示适用计算机架构。通常修订版本号是由 Ubuntu 开发者或创建这个软件包的人指定。在软件包被修改过之后, 修订版本号加 1。

以 mouseemu_0.16-0ubuntu3_i386.deb 软件包为例, mouseemu 是软件包名, 0.16 是此软件包的版本号, 0ubuntu3 是修订版本号, i386 是指此软件包适用的计算机架构为 i386。

3. 软件包的优先级

Ubuntu Linux 为每个软件包指定了一个优先级, 作为软件包管理器选择安装和卸载的一个依据。表 11-1 列出了 Ubuntu 定义的所有软件包优先级描述。

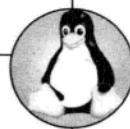


表 11-1 软件包优先级描述

级别	含义	补充说明
Required (必需)	该级别软件包是保证系统正常运行所必需的 若缺少该级别软件包，系统会运行困难或不好操作 该级别软件包是任何Linux系统的标准件 该级别软件包是否安装不影响系统的正常运行 该级别软件包可能与其他高级别软件包存在冲突	包含所有必要的系统工具。尽管 Required 级别的软件不能满足整个系统的服务，但至少能够保证系统正常启动。如果删除其中一个软件包，系统将受到损坏而无法恢复。系统正常运行所必需的软件包。这些软件包不能被移除，否则系统可能会崩溃，一个只包含必需软件包的系统也许无法正常使用，但是它为系统管理员启动系统安装其他软件提供了足够的功能。例如，bash、mount、upstart
Important (重要)		任何在类 UNIX 系统都会有的程序被划定为重要软件包，此类包构成基本系统，这类包会影响其他软件包的运行。该级别软件包是一些实现系统底层功能的程序。例如，aptitude、ubuntu-keyring、cpio
Standard (基本)		该级别的软件包（以及上面两个级别的软件包）组成了 Ubuntu 的最小化安装。不包括大型应用软件。该级别的软件包支撑命令行控制台系统运行，通常作为默认安装选项，例如，memtest86、telnet、pppconfig、ed
Optional (可选)		该级别软件包，包含了所有的用于想要安装的文件，即使对它们并不熟悉。包括 X 窗口、完整的 TeX 发布版，以及许多应用程序。注意可选软件包彼此间不应存在冲突。这包括 X11，所有的 TeX 和许多应用程序。该级别的软件包用于满足用户特定的需求或服务，它们不会影响系统的正常运行
Extra (额外)		这一层是用户级的应用程序级别，用户可以自行安装的，可以从网上下载，符合自己的工作、学习要求。如 Firefox、pidgin 等

在 Ubuntu 系统中规定，任何高优先级的软件包都不能依赖于低优先级的软件包。这样可以实现按照优先级一层层冻结系统。在新版本发布准备阶段，优先级的作用就显得更为重要。

Ubuntu 的基本系统由 Required 级和 Important 级软件包组成。发布的时候，属于这两类优先级的软件包首先被冻结。由于这些软件包是其他软件包所依赖的，它们能保证整个系统的稳定，这是 Ubuntu 发布新版本所必需的。然后，冻结 Standard 级软件包，紧接着，在发布新版本之前对 Optional 级和 Extra 级软件包进行冻结。

4. 软件包的状态

在使用系统过程中，用户不断地安装、卸载软件包。为了记录用户的安装行为，Ubuntu 对软件包定义了两种状态，如下所示。

- 期望状态：标记用户希望将某个软件包处于的状态。
- 当前状态：标记用户操作该软件包后的最终状态。

在现行版本的 Ubuntu Linux 中，检测的功能都由 Ubuntu 系统自动完成，判断用户是否可以安装，是否可以升级，这样有利于用户方便使用系统，并且给自动更新软件提供了数据依据。

使用 dpkg 命令可以查看软件包的状态，如下所示：

```
*@*:~$ dpkg -l | more
期望状态=未知(u)/安装(i)/删除(r)/清除(p)/保持(h)
| 当前状态=未(n)/已安装(i)/仅存配置(c)/仅解压缩(u)/配置失败(f)/不完全安装(h)/触
发器等待(w)/触发器未决(t)
| / 错误?=(无)/保持(h)/须重装(r)/两者兼有(x) (状态, 错误: 大写=故障)
|| / 名称 版本
+++-----
```

```
=====
ii a2ps                               1:4.14-1
  GNU a2ps - 'Anything to PostScript' converter and pretty-printer
...
ii acl                                2.2.47-2
```

每条记录对应一个软件包，注意每条记录的第一、二、三个字符。这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。

- 第一字符为期望值，其包括如下所示 5 种状态标识符号：

- (1) u 表示状态未知，这意味着软件包未安装，并且用户也未发出安装请求。
- (2) i 表示用户请求安装软件包。
- (3) r 表示用户请求卸载软件包。
- (4) p 表示用户请求清除软件包。
- (5) h 表示用户请求保持软件包版本锁定。

- 第二列，是软件包的当前状态。此列包括软件包的六种状态如下所示：

- (1) n 表示软件包未安装。
- (2) i 表示软件包安装并完成配置。
- (3) c 表示软件包以前安装过，现在删除了，但是它的配置文档还留在系统中。
- (4) u 表示软件包被解包，但还未配置。
- (5) f 表示试图配置软件包，但是失败了。
- (6) h 表示软件包安装，但是没有成功。

- 第三列标识错误状态，其包括四种状态。第一种状态表示没有问题，标识为空。其他三种状态标识符号所标识的相应问题如下所示：

- (1) h 软件包被强制保持，因为有其他软件包依赖需求，无法升级。
- (2) 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）。
- (3) x 软件包被破坏，并且被强制保持。

5. 软件包的依赖性关系

Linux 操作系统是一个复杂的系统，包含了大量的软件组件。如果各个组件之间都是相互独立的话，这样系统是难以使用的，而且是不可以使用的。组件之间没有通信，没有数据的交换，没有共享的接口，难以组成一个有机的整体，也不可以使系统有效地运行。若每个组件之间没有太多的共同点，那也对系统的运行不利。系统不能有效地使用组件，组件并不能充分体现各自的性能。

因此，若要求它们能够成为一个有机整体，支撑 Linux 系统的正常运转，就必须要求各个组件密切配合。这就是 Linux 操作系统最初的设计理念——尽可能提高软件系统内部的耦合度。换句话说，某个软件组件是否能够正常运行或运行得更好，依赖于其他一些软件组件的存在。这样做好处是使系统更加致密、紧凑，减少中间环节可能引发的错误。然而，随即带来两个负面问题，软件组件依赖和软件组件冲突问题。

为了解决这个问题，Debian 提出了程序依赖性机制，并做了详细的定义。程序依赖性是用来描述独立运行程序与当前系统中程序之间存在的关联程度。表 11-2 列出了 Ubuntu 中依赖性关系的定义。



表 11-2 依赖性关系定义

依 赖 关 系	关 系 描 述
依赖 (depends)	要运行软件包 A 必须安装软件包 B，甚至还依赖于 B 的特定版本，通常版本依赖有最低版本限制
推荐 (recommends)	软件包维护者认为所有用户都不会喜欢缺少软件包 A 的某些功能，而这些功能需要 B 来提供
建议 (suggests)	软件包 B 能够增强软件包 A 的功能
替换 (replaces)	软件包 B 安装的文件被软件包 A 中的文件删除或覆盖了
冲突 (conflicts)	如果系统中安装了软件包 B，那么软件包 A 将无法运行。“Conflicts”常和“Replaces”同时出现
提供 (provides)	软件包 A 中包含了软件包 B 中的所有文件和功能

软件包管理器将依据软件包的“依赖关系”完成相关组件的安装或卸载。例如，包含 GNU C 编译器的软件包依赖于包含链接器和汇编器的 binutils 软件包。如果用户试图在没有安装 binutils 的情况下安装 GCC，软件包管理器将会中止安装，并出现错误提示，要求用户必须事先安装 binutils。

11.1.2 RPM 软件包

RPM 是 RedHat Package Manager 的缩写，本意是 Red Hat 软件包管理，顾名思义是 Red Hat 贡献出来的软件包管理。RPM 有两层意思：一种软件包的格式和一种管理软件包的管理软件。开始的时候 RPM 是 Red Hat 公司为 Red Hat Linux 开发的一种软件包，参照了 Debian 的 Deb 软件包；而软件包管理工具 RPM 则如同 Debian 中的 dpkg 工具。目前 rpm 广泛用于 Linux 的其他发行版本和一些其他的操作系统，如 Novell NetWare 6.5 SP3 以上的版本，IBM 的 AIX 4 以上的版本。

RPM 的出现无疑减轻了用户的负担，最大限度地方便了用户，它提供了众多的功能使得维护 Linux 系统要比以前简单得多。安装、卸装和升级 RPM 软件包，只需一条命令就可以完成，而且还提供了图形操作界面。RPM 还维护了一个所有已安装的软件包和文件的数据库，可以让用户进行功能强大的软件包查询和验证工作。同时，在软件包升级过程中，RPM 会对配置文件进行特别处理，因此用户绝对不会丢失以往的配置信息，这对于直接使用.tar.gz 文件是不可能实现的，因为解开.tar.gz 文件后将覆盖掉用户原来的配置信息。

同样，RPM 的使用，也为程序员简化了软件包的发布工作。使用 RPM，程序员可以将软件的源代码和二进制软件包一起打包成 rpm 软件包，并发布 rpm 软件包。制作 rpm 软件包，操作简单，只需把一个主文件和可选的补丁程序打包。

1. rpm 软件包的命名

RPM 对 rpm 软件包的命名并没有硬性的规定。然而，一般情况下 rpm 软件包都会采用如同 Deb 的软件命名一样的命名方式，rpm 软件包也有它的软件包标签，表示软件包的发布。rpm 软件包的标签需要包括以下信息：

- 软件包的名称。
- 软件的版本（原软件的源代码的版本）。
- 软件包的发布版本（描述软件包被重新构建的次数，使用同样的软件版本），这个字段还被用于表示此软件包特定的 Linux 发布版本，如 mdk/mdv 表示用于 Mandriva

Linux; fc4 表示用于 Fedora Core 4; rhl9 表示用于 Red Hat Linux。

- 系统体系结构（描述软件包应用系统的体系结构，如 i386、i686、ppc、athlon 等）。所以，一般 rpm 软件包的标签命名为：

<软件包名>-<软件版本>-<发布版本>.<体系结构>.rpm

例如：

eva-0.4.9-bugfix73_1.fc10.i386.rpm

fc10 表示此 rpm 软件包用于 Fedora Core 10 系统中，i386 表示体系结构用于为 i386 系列的计算机上。

2. RPM 包的优点

Linux 提供了软件包管理功能，它为系统维护和升级带来了方便。RPM 具有其他软件包管理工具所不具有的特色，因此深受用户的喜爱。它的优点包括：模块升级性、强大的软件包信息查询功能、系统验证、保持软件包原始特性。

3. RPM 的不足

RPM 软件包管理工具也不是十分完美的，它也存在一些不足，如下所示：

- 无法自动解决依赖问题：RPM 会告诉你缺什么，但不能自己去获取、安装所缺少的包。但这个主要的不足，基本上可以由一些其他工具来弥补。
- 没有交互式的配置：软件包通常都包含配置文件和安装后可运行的脚本，但这种配置只能在安装后单独进行。
- 难以预防包中的错误：每个人都会犯错误，包维护者也不例外。安装了说明文件有误的包，可能导致无法运行程序。
- 缺乏兼容性保证：有很多 Linux 发行商和独立的个人在使用 RPM 。

11.1.3 源码包

顾名思义，源码包就是源代码可见的软件包。目前基于 Linux 和 BSD 系统的软件是最常见的源码包；在国内开源的软件几乎绝迹；大多开源软件都是国外出品；在国内较为出名的开源软件有 fcitx、Lumaqq 及 scim 等。

使用 tar 程序归档文件，生成的文档一般都比较庞大，需要对这一文件进行适当的压缩，这就需要使用到压缩软件了。比较常用的压缩程序包括 gzip 和 bunzip2。使用 gzip 对 tar 文件进行压缩生成后缀为 tar.gz 或 gz；而如用 bunzip2 程序压缩的 tar 文件，会使用后缀 tar.bz2 或 bz2。

用 tar 程序打包并用 gzip 程序压缩的软件包，安装时使用终端安装。要安装这种软件包，需要先对软件包进行解压缩，使用 “tar -zxfv filename.tar.gz” 命令可以对软件包进行解压缩，解压缩所得的文件在以 Filename 为名的目录中。进入该目录，可以看到解压缩出来的文件。安装的过程普遍有 3 个步骤：进入 Filename 目录运行 ./configure 执行文件，配置并生成 Makefile 文件；使用 make 命令完成任务，执行 Makefile 文件的配置，编译软件包；使用 make install 命令完成软件包的安装操作。使用各种软件都有不同的安装方法，但是一般每个软件包解压缩后都会包含有 Install 和 Readme 文件，在这两个文件中会有详细的安装指导。

用 tar 程序打包并用 bunzip2 程序进行压缩的软件包，安装方法和使用 gzip 压缩的软件包



相同。它的优点是压缩率非常高，需要使用命令“`bunzip2 filename.tar.bz2`”进行解压。

无论是`.tar.gz`还是`.tar.bz2`软件包，它的安装都不适合普通的使用者。源码包的安装比较复杂，没有相关的环境是安装不成的。也就是说，源码包并没有提供软件的相关性信息。安装源码包之后，用户删除此软件一定需要知道它安装在了哪个位置，把整个文件夹删除就可以，但这对普通用户来说，也是比较艰巨的事情。我们可以通过查看`Makefile`文件了解软件的安装路径和配置信息。建议初学者还是使用软件包管理工具来维护软件包，完成软件包的安装、删除和升级等操作。对自己喜欢的一些软件也可以在网上寻找相应版本的`Deb`软件包，这样安装会很顺利。

技术提示

`tar`是一种 UNIX 和类 UNIX 系统下的打包工具，它把多个文件合并为一个文件。最初设计此程序是为了将文件备份到磁带上（tape archive），因而得名`tar`。经`tar`打包的文件，文件的后缀为“`.tar`”。目前，`tar`文件格式已经成为 POSIX 标准，最初是 POSIX.1-1988，目前是 POSIX.1-2001。常用的`tar`程序是自由软件基金会开发的 GNU 版，目前的稳定版本是 1.22，发布于 2009 年 3 月 5 日。

11.1.4 虚拟软件包

在 Ubuntu Linux 中还有一个软件包的概念，就是虚拟软件包。虚拟软件包是一个统称，它代指一组具有相近功能的软件包中的任何一个。例如，`tin` 和 `trn` 都是新闻组阅读软件，当系统中某个文件的打开需要使用新闻阅读器时，它们中的任何一个都可以满足要求。因此，这两个软件包一起组成了一个叫做`news-reader`（新闻阅读器）的“虚拟软件包”。

类似的，许多`exim`、`exim4`、`sendmail`、`postfix`这样的软件包都提供邮件传输代理的功能。因此，它们一起组成了一个称为`mail-transport-agent`（邮件传输代理）的虚拟包。安装了它们中的任何一个，都会满足其他依赖于邮件传输代理功能的服务需要。

Ubuntu 有如此的机制，如果系统中安装了同一虚拟包的多个软件包，系统管理员需要指定其中之一作为首选软件。

11.2 字符界面软件包管理工具



Ubuntu 集成了比较多的字符界面软件包管理工具，如`dpkg`、`apt-get`、`aptitude`、`dselect`和`tasksel`。这几个软件包管理工具不是为了相互代替才产生的。它们之间有着不少的共同点，也可以彼此利用。软件包管理工具`dpkg`是 Debian 包的安装工具；`apt-get`是 APT 的命令行前端；`aptitude`是 APT 的高级的字符界面和命令行前端；`dselect`是使用菜单界面的包管理工具；`tasksel`是 Task 安装工具。

11.2.1 dpkg 软件包管理工具

`dpkg`是 Debian 软件包管理器的基础，由伊恩·杰克逊创建于 1993 年。`dpkg`用于安装、

卸载软件包，并提供.deb 软件包相关的信息。dpkg 是 Ubuntu 软件包管理系统的中流砥柱，负责安装、卸载软件包、配置和维护已安装的软件包，也是 Ubuntu 系统中众多软件包管理工具的后端。

dpkg 通过数据库来维护系统中的软件，这包括文档清单、依赖关系、软件状态等详细的内容，通常数据库保存在/var/lib/dpkg 目录下。同时 dpkg 确保系统和数据库的情况相一致。因此，dpkg 在处理程序时很迅速且很完美。

dpkg 软件包管理工具是个庞大的家族，不但提供了大量的参数选项，同时也提供了许多子命令比如:dpkg-deb; dpkg-divert; dpkg-query; dpkg-split; dpkg-statoverride; start-stop-daemon 等。在使用 dpkg 工具之前，我们先了解它的相关用法，如例 11.1 所示。

例 11.1：使用 help 命令

使用 help 命令查看 dpkg 软件包管理工具的用法，执行命令如下：

```
dranyu@dranyu-desktop:~/software$ dpkg -- help
用法: dpkg [<选项> .....] <命令>
命令:
 -i|--install      <.deb 文件名> ..... | -R|--recursive <目录> .....
 --unpack          <.deb 文件名> ..... | -R|--recursive <目录> .....
 -A|--record-avail <.deb 文件名> ..... | -R|--recursive <目录> .....
 - --configure      <软件包名> ..... | -a|--pending
 - --triggers-only <软件包名> ..... | -a|--pending
 -r|--remove       <软件包名> ..... | -a|--pending
 -P|--purge        <软件包名> ..... | -a|--pending
 .....
可供--compare-version 使用的比较运算符有:
 lt le eq ne ge gt      (如果版本号为空, 那么就认为它先于任意版本号);
 lt-nl le-nl eq-nl gt-nl (如果版本号为空, 那么就认为它后于任意版本号);
 < << <= = > >      (仅仅是为了与主控文件的语法兼容)。
```

下面将以 LinuxQQ 软件包和 g++ 软件包作为命令 dpkg 的操作对象说明 dpkg 的使用，以及 dpkg 系列中的其他命令的操作。

补充说明

dpkg 由以下几个原始的辅助程序演化而来：

- (1) dpkg-deb：操作.deb 文件。
- (2) dpkg-ftp：一个旧的软件包获取命令。
- (3) dpkg-mountable：一个旧的软件包获取命令。
- (4) dpkg-split：将大软件包分割成小软件包。

1. 显示软件包的相关信息

dpkg-deb：用于处理本地 Ubuntu 软件包，能够从.deb 文档中提取软件包信息和数据。

例 11.2：显示软件包的信息

用户可以使用 dpkg-deb -I packagefileName 或 dpkg -I packagefileName 来完成查看软件包信息的操作，命令执行和结果如下所示：

```
dranyu@dranyu-desktop:~/software/qq$ dpkg -I linuxqq_v1.0.2-beta1_i386.deb
新格式的 debian 软件包，格式版本 2.0
```



```
大小 5028066 字节: 主控包=442 字节
    0 字节,     0 行      conffiles
  412 字节,    11 行      control
Package: linuxqq
Priority: extra
Section: Internet
Installed-Size: 6264
Maintainer: Tencent Inc.
Architecture: i386
Version: v1.0.2-beta1
Depends: gtk2.0-binver-2.10.0, libc6, libcairo2, libglib2.0-0, libgtk2.0-0,
libpango1.0-0
Description: QQ - The most popular free instant messaging program in China.
 * Chat with more than 590,000,000 people all over the world.
 * Free to download or distribute absolutely
```

其中“格式版本 2.0”表示此软件包使用的 Debian 软件包格式，“Linuxqq”是此软件包的名称，而“version:v1.0.2-beta1”表示此软件包的版本是 v1.0.2-beta1 版本，“Architecture”描述了此软件包用于的体系结构是 i386，“Depends”描述了此软件包的依赖关系，“Description”描述的是维护者的信息，优先级和相应的软件包的描述。使用 dpkg 选项时注意区别大小写，此处使用的是大写的 I。

例 11.3：列出软件包的内容

使用 dpkg-deb -c packageFileName 或 dpkg -c packageFileName 可以完成任务，命令执行与结果如下所示：

```
dranyu@dranyu-desktop:~/software/qq$ dpkg -c linuxqq_v1.0.2-beta1_i386.deb
drwxr-xr-x root/root          0 2009-01-04 13:10 .
drwxr-xr-x root/root          0 2009-01-04 13:10 ./usr/
drwxr-xr-x root/root          0 2009-01-04 13:10 ./usr/bin/
...
-rw-rw-rwx root/root       11 2008-12-07 13:43 ./usr/share/doc/linuxqq/qq
```

此处罗列出了 LinuxQQ 软件包的所有数据，安装的时候，系统会把这些文件复制到相应的目录下，如目录./usr/下的文件就会复制到系统的/usr/目录下。

2. 软件包的数据库查询

目前版本的 dpkg 使用文本文档作为数据库。一般来说，系统都会把数据保存到 /var/lib/dpkg 目录下。通常在 status 文档中存储软件状态和控制信息。在 /var/lib/dpkg/info/ 目录下备份控制文档，并在其下的.list 文档中记录安装文档清单，其下的.mdasums 保存文档的 MD5 编码。

例 11.4：罗列数据库中软件包的情况

使用 dpkg -l 命令可以罗列目前系统中安装的软件数目，以及记录在数据库中的软件的安装情况，命令如下所示：

```
dranyu@dranyu-desktop:~/software$ dpkg -l | more
期望状态=未知(u)/安装(i)/删除(r)/清除(p)/保持(h)
| 当前状态=未(n)/已安装(i)/仅存配置(c)/仅解压缩(U)/配置失败(F)/不完全安装(H)/触发器等待(W)/触发器未决(T)
| / 错误?=(无)/保持(H)/须重装(R)/两者兼有(X) (状态, 错误: 大写=故障)
|| 名称                                版本                                简介
+---+
```

```
=====
ii a2ps                               1:4.14-1           GNU
a2ps - 'Anything to PostSc
ript' converter and pretty-printer
ii abiword                            2.6.6-0ubuntu1
efficient, featureful word pro
cessor with collaboration
ii abiword-common                     2.6.6-0ubuntu1
efficient, featureful word pro
cessor with collaboration -- commo
ii abiword-help                       2.6.6-0ubuntu1
online help for AbiWord
ii abiword-plugin-grammar            2.6.6-0ubuntu1
grammar checking plugin for Abi
iWord
ii abiword-plugin-mathview           2.6.6-0ubuntu1
equation editor plugin for Abi
Word
ii abiword-plugins                   2.6.6-0ubuntu1
transitional plugins package for
AbiWord
ii acl                                2.2.47-2           Access
control list utilities ...
...
```

系统罗列出软件包的相关信息：记录的第一个字段表示安装状态，如 ii；第二字段表示软件包的名称，如 acl；第三字段表示软件包的本版，如 2.2.45-2；以及其他。在配合管道命令“|”的情况下，就可以查询特定的软件包的状态信息，如命令 `dpkg -l | grep Linuxqq`。

例 11.5：查询目录的软件包数据库中软件包 g++ 的状态信息

使用匹配的字符来对软件包数据库进行模糊查询，执行如下命令：

```
dranyu@dranyu-desktop:~$ dpkg -l g++
期望状态=未知(u)/安装(i)/删除(r)/清除(p)/保持(h)
! 当前状态=未(n)/已安装(i)/仅存配置(c)/仅解压缩(U)/配置失败(F)/不完全安装(H)/触发器等待(W)/触
发器未决(T)
! / 错误?=(无)/保持(H)/须重装(R)/两者兼有(X) (状态, 错误: 大写=故障)
! | 名称      版本      简介
+---+
ii g++       4:4.3.3-lubuntu   The GNU C++ compiler
...
```

第一个字符 i 表示了期望状态为安装，第二个字符 i 表示当前软件包的状态为已安装，第三个字符为空表示没有错误。

假如觉得 `dpkg` 的参数过多，不利于记忆的话，还可使用 `dpkg-query` 进行 `dpkg` 数据库查询。

对比两种数据库查询命令如下：

- (1) 查询系统中属于 g++ 的文档： \$ `dpkg -l` `g++` 或 \$ `dpkg-query -L g++`
- (2) 查看软件 g++ 的周详信息： \$ `dpkg -s g++` 或 \$ `dpkg-query -s g++`
- (3) 查看系统中软件包状态，支持模糊查询： `dpkg -l` 或 \$ `dpkg-query -l`
- (4) 查看某个文档的归属包： \$ `dpkg-query -S g++` 或 \$ `dpkg -S g++`

3. 安装软件包

我们可以从上边的 `dpkg -l` 命令查看到，数据库中并没有 LinuxQQ 软件包的相关信息。



我们可以使用命令 `dpkg -l | grep *linuxqq*` 来判断是否安装了 LinuxQQ 软件。如果没有任何结果则表示数据库中并没有 linuxqq 的相关信息。

例 11.6：安装 Linuxqq 软件包

假设目前的工作目录为存放 LinuxQQ 软件包的目录，LinuxQQ 软件包的文件名为：`linuxqq_v1.0.2-beta1_i386.deb`。安装此软件包的命令如下所示：

```
dranyu@dranyu-desktop:~/software/qq$ sudo dpkg -i linuxqq_v1.0.2-beta1_i386.deb  
[sudo] password for dranyu:  
选中了曾被取消选择的软件包 linuxqq。  
(正在读取数据库 ... 系统当前总共安装有 161189 个文件和目录。)  
正在解压缩 linuxqq (从 linuxqq_v1.0.2-beta1_i386.deb) ...  
正在设置 linuxqq (v1.0.2-beta1) ...
```

执行 `sudo dpkg -i linuxqq_v1.0.2-beta1_i386.deb` 命令后，`dpkg` 就会将 LinuxQQ 软件包安装到 Ubuntu 系统中。

例 11.7：查看 LinuxQQ 软件包是否已安装成功

查看 LinuxQQ 安装情况的命令如下所示：

```
dranyu@dranyu-desktop:~/software/qq$ dpkg -l linuxqq  
期望状态=未知(u)/安装(i)/删除(r)/清除(p)/保持(h)  
| 当前状态=未(n)/已安装(i)/仅存配置(c)/仅解压缩(U)/配置失败(F)/不完全安装(H)/触发器等待(W)/触发器未决(T)  
| / 错误?=(无)/保持(H)/须重装(R)/两者兼有(X) (状态, 错误: 大写=故障)  
|| 名称      版本      简介  
+---  
ii  linuxqq  v1.0.2-beta1  QQ - The most popular free instant messaging program in China.
```

我们可以发现记录 LinuxQQ 的软件包状态显示为 `ii`，表示 LinuxQQ 已成功安装。在终端下输入 `QQ`，就可以运行我们熟悉的 `QQ` 登录界面，输入你的 `QQ` 账号和密码，就可以使用 `QQ` 联系 `QQ` 好友了。

例 11.8：详细解释 LinuxQQ 软件包的安装过程

其实在软件包的安装过程中，`dpkg` 工具执行了两个阶段的步骤：首先解包；然后运行 `postinst` 控制脚本（假如有的话），这就是所谓的配置阶段。当完成软件包解包后，`dpkg` 的`--install` 选项自动调用配置阶段。注意，操作时应当指定 `.deb` 文档的实际路径，仅仅给出文档名无法找到对应文档。

这样，软件的安装就可被拆分为两个对立的过程：`dpkg -unpack`；`dpkg -configure`。我们可以按照这种方式安装 LinuxQQ，也就是执行次命令。第一次执行 `dpkg -unpack linuxqq_v1.0.2-beta1_i386.deb`，第二次执行 `dpkg -configure linuxqq_v1.0.2-beta1_i386.deb` 命令。为了可以详细了解 `dpkg` 的安装过程，我们来细说安装的步骤：

第一步：在终端输入命令如下：

```
dranyu@dranyu-desktop:~/software/qq$ sudo dpkg --unpack linuxqq_v1.0.2-beta1_i386.deb
```

为了可以查看到目前 LinuxQQ 软件包的安装情况，我们输入如下命令，并按下回车键执行命令：

```
dranyu@dranyu-desktop:~/software/qq$ dpkg -l linuxqq  
期望状态=未知(u)/安装(i)/删除(r)/清除(p)/保持(h)  
| 当前状态=未(n)/已安装(i)/仅存配置(c)/仅解压缩(U)/配置失败(F)/不完全安装(H)/触发器等待(W)/触发器未决(T)  
| / 错误?=(无)/保持(H)/须重装(R)/两者兼有(X) (状态, 错误: 大写=故障)
```

名称	版本	简介
----	----	----

```
iu linuxqq v1.0.2-beta1 QQ - The most popular free instant messaging program in China.
```

说明此时系统中已经解压了 LinuxQQ 软件包，但是未经配置，也就是我们的安装工作还没有完成。

解包可能包括以下步骤：将控制文档解压到临时目录；运行 `preinst`（假如有的话）。通常为停止相关服务；解压配置文档到 `/etc` 目录，并添加 `.dpkg-new` 后缀，以防止冲突；解压 `data.tar.gz` 到系统根目录（`--root=dir`）；将控制文档解压到 `/var/lib/dpkg/info` 目录并添加软件名前缀，对数据库进行更新；数据库中将软件包的状态配置为“`unpacked`”。仅仅解包，并不能确保软件能够正常运行。

第二步，接下来我们就要对软件包进行配置操作，输入配置命令，并按回车键确认完成输入执行命令：

```
dranyu@dranyu-desktop:~/software/qq$ sudo dpkg -configure linuxqq
```

这样 LinuxQQ 安装完毕，通过命令：

```
dranyu@dranyu-desktop:~/software/qq$ dpkg -l linuxqq
```

```
ii linuxqq v1.0.2-beta1 QQ - The most popular free instant messaging program in China.
```

可查看到 LinuxQQ 已安装成功。

软件解包后，还需要对其进行调整，比如根据实际情况对配置文档进行修改。这一步通常是在安装过程中自动进行的，当然能够手动调用，这次用到的是软件包名，而不是 deb 文档名。通过命令 `dpkg -Llinuxqq` 能够得以验证，这一过程可能包括以下步骤：询问对配置文档的处理方法，覆盖当前版本，还是不做处理（当软件升级时会碰到这种情况）；运行 `postinst` 脚本；将软件包标记为“`installed`”。

假如想再次修改配置选项，可执行如下命令：

```
#dpkg -reconfigure linuxqq
```

解压之后，我们运行的命令中都是使用 LinuxQQ 作为 `dpkg` 命令的参数而不是文档名。

4. 卸载软件包

在 Ubuntu 中卸载和清除软件包是两个不同的概念。不同之处在于软件包被删除（卸载）后，软件包的配置文档仍会留在系统中，只有清除时才会删除它们。默认情况下，Ubuntu 仅会做删除操作，除非您明确指出才会将配置文档删除。假如要清除软件包，则在清除前将会隐含地执行删除操作。

要删除一个软件包，与 `dpkg` 需要使用 `--remove` 选项将软件包卸载和安装不同，删除只需要软件程序名，而不是实际的 deb 文档名。

例 11.9：删除 LinuxQQ 软件包

在不需要使用 LinuxQQ 软件的时候，我们可删除 LinuxQQ 软件包，使用命令如下所示：

```
dranyu@dranyu-desktop:~/software/qq$ sudo dpkg -r linuxqq
```

```
[sudo] password for dranyu:
```

```
(正在读取数据库 ... 系统当前总共安装有 161197 个文件和目录。)
```

```
正在删除 linuxqq ...
```

例 11.10：查看执行删除操作后 LinuxQQ 软件包的状态信息

使用 `dpkg -l` 命令查看 LinuxQQ 是否被删除，执行命令与结果显示如下：

```
dranyu@dranyu-desktop:~/software/qq$ dpkg -l linuxqq
```

```
期望状态=未知(u)/安装(i)/删除(r)/清除(p)/保持(h)
```



```
| 当前状态=未(n)/已安装(i)/仅存配置(c)/仅解压缩(U)/配置失败(F)/不完全安装(H)/触发器等待(W)/触发器未决(T)
| / 错误?=(无)/保持(H)/须重装(R)/两者兼有(X) (状态, 错误: 大写=故障)
|| | 名称 版本 简介
+---+
rn linuxqq v1.0.2-beta1 QQ - The most popular free instant messaging program in
China.
```

我们可以发现 LinuxQQ 记录行的软件状态标志显示为 `rn`, 表示 LinuxQQ 软件包已经被卸装。

`dpkg` 命令卸装软件包的详细过程可分为以下几个步骤: 首先运行 `prerm` 脚本(假如有的话); 然后保留配置文档, 其他文档全部删除; 删掉 `/var/lib/dpkg/info` 目录下软件包的任何配置文档, 保留 `postrm` 和 `list` 文档; 最后, `dpkg` 在数据库中将软件状态修改为删除。

如果要把软件包的配置文件也删除掉, `dpkg` 提供了清除操作。

例 11.11: 清除 LinuxQQ 软件包

要清除 LinuxQQ 软件包, 可执行命令如下所示:

```
dranyu@dranyu-desktop:~/software/qq$ sudo dpkg -P linuxqq
(正在读取数据库 ... 系统当前总共安装有 161197 个文件和目录。)
正在删除 linuxqq ...
```

消除操作的过程为: 删掉软件包中的任何文档, 任何的文档都被解除了; 运行 `postrm` 脚本, 假如还存在; 删掉 `postrm` 和 `list` 文档保留的两个文档, 从系统中解除 `postfix.postrm` 和 `postfix.files`, 在 `dpkg` 的数据库中将软件包标记为未安装。

例 11.12: 查看执行清除操作后 LinuxQQ 软件包的状态信息

在清除 LinuxQQ 软件之后, 我们可查看清除软件包后软件包的状态, 使用命令如下所示:

```
dranyu@dranyu-desktop:~/software/qq$ dpkg -l linuxqq
期望状态=未知(u)/安装(i)/删除(r)/清除(p)/保持(h)
| 当前状态=未(n)/已安装(i)/仅存配置(c)/仅解压缩
(U)/配置失败(F)/不完全安装(H)/触发器等待(W)/触发器未决(T)
| / 错误?=(无)/保持(H)/须重装(R)/两者兼有(X) (状态, 错误: 大写=故障)
|| | 名称 版本 简介
+---+
pn linuxqq <无> (无相关介绍)
```

我们可发现 LinuxQQ 软件包的状态标识为 `pn`, 这表示 LinuxQQ 软件包的相关文档都从系统中被删除了。

11.2.2 apt-get 软件包管理工具

APT 是 Debian 的 deb 软件包高级管理工具, 由一组以 `apt` 开头的命令组成, 如 `apt-get`、`apt-cache` 和 `apt-setup` 等。它的最低层还是调用 `dpkg` 程序来处理软件包。通过 `apt-get` 工具可自动解除软件包的依赖关系, 方便软件的安装和升级。它是 Debian 发行版的最大特色。Ubuntu 沿用了 Debian 的这一 APT 机制。

`apt-get` 命令是 APT 机制中的一个强大的命令行工具, 用于同 Ubuntu 的 Advanced Packaging Tool (APT) 一起执行诸如安装新软件包、升级已有软件包、更新包列表索引, 甚至是升级整个 Ubuntu 系统等功能。作为一个简单的命令行工具, `apt-get` 对于服务器管理员来

说比 Ubuntu 中的其他软件包管理工具有相当多的优点。这些优点包括便于在简单终端链接(SSH)中使用，同时能够用于系统管理脚本中，以便能被 cron 动作计划工具自动运行。

apt-get 是使用比较频繁的一个软件包管理工具，执行速度快，工作效率高，相关的配置文件都保存在/etc/apt/目录下。

1. 配置软件源

在/etc/apt/目录中比较重要的一个文件是 sources.list 文件，里边的内容为软件源的路径。使用 vi source.list 命令可以在/etc/apt/目录下打开 source.list 文件。source.list 中的内容有一定的格式为：

```
deb [web 或 ftp 地址] [发行版名字] [main/contrib/non-free]
```

以下是 Ubuntu 9.04 发行版本下的 sources.list 文件的片段：

```
deb cdrom:[Ubuntu 9.04 _Jaunty Jackalope_ - Release i386 (20090420.1)]/ jaunty main
restricted
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
...
deb http://cn.archive.ubuntu.com/ubuntu/ jaunty main restricted
deb-src http://cn.archive.ubuntu.com/ubuntu/ jaunty main restricted
...
## Major bug fix updates produced after the final release of the
## distribution.
deb http://cn.archive.ubuntu.com/ubuntu/ jaunty-updates main restricted
deb-src http://cn.archive.ubuntu.com/ubuntu/ jaunty-updates main restricted
```

第一行说明软件源来自光盘存储器。deb 开头的行设置下载软件包的路径，而以 deb-scr 开头的行设置下载软件包源码的路径。

Ubuntu 发行的光盘映像文件本来就是一个软件库，设置光盘作为软件源也就成为必然，也是很重要的，有一部分软件都可以从发行光盘中找到。我们可以使用 apt-cdrom 命令添加光盘软件源到 apt 的软件源文件中。

配合着 add 命令我们就可以添加 cdrom 到软件源中。如果我们没有刻录光盘，而是把光盘文件下载到硬盘中，我们同样可以设置硬盘上的映像文件作为我们的软件源。

第一步，把映像文件挂载到系统的一个目录上，假设我们需要把光盘映像文件挂载到 /mnt/diskiso，我们先在/mnt/目录下创建一个目录 diskiso，进入/mnt/目录，在终端使用：

```
dranyu@dranyu-desktop:/mnt$ sudo mkdir diskiso
```

创建一个新文件夹 diskiso，我们要把映像文件挂载到此文件中，还需使用挂载命令 mount。

第二步，挂载映像文件。

```
dranyu@dranyu-desktop:~/software$ sudo mount -o loop ubuntu-9.04-desktop-i386.iso
/mnt/diskiso
```

执行命令完毕之后，需要把这个链接写入到 sources.list 中。

第三步，使用编辑器 vi 或是其他编辑器打开 sources.list，添加如下代码：

```
deb file:///mnt/diskiso jaunty main restricted
```

完成了硬盘上映射文件的添加，使用硬盘上的目录作为安装源的优势使安装速度快很多。安装一个 GNOME 系统只要几分钟时间，如果上网下载可能要几十分钟，这提高了系统的工作效率，为用户节省了时间。



2. 更新软件源

修改 sources.list 文件之后都需要执行 apt-get update 命令更新软件源，命令如下所示：

```
dranyu@dranyu-desktop:/mnt$ sudo apt-get update
忽略 cdrom://Ubuntu 9.04 _Jaunty Jackalope_ - Release i386 (20090420.1) jaunty/main
Translation-zh_CN
忽略 cdrom://Ubuntu 9.04 _Jaunty Jackalope_ - Release i386 (20090420.1) jaunty/restricted
Translation-zh_CN
命中 http://cn.archive.ubuntu.com jaunty Release.gpg
命中 http://cn.archive.ubuntu.com jaunty/main Translation-zh_CN
.....
```

上述显示结果中，“忽略”字符串表示此软件源没有更新，“命中”字符串则表示此链接的软件源得到了更新。

3. 查找软件包

在软件源中查找用户需要的软件包，使用 apt-cache 命令完成任务。用户只要知道软件包的名称，或是软件包的用户，就可以在软件源中查找所需要的软件包。

例 11.13：搜索软件包

查找软件包信息中包括字符串“tomcat”的软件包

```
dranyu@dranyu-desktop:/etc$ apt-cache search tomcat
libservlet2.3-java - Servlet 2.3 and JSP 1.2 Java classes and documentation
...
tomcat5.5 - Servlet and JSP engine
tomcat5.5-admin - Java Servlet engine -- admin & manager web interfaces
tomcat5.5-webapps - Java Servlet engine -- documentation and example web applications
...
tomcat6 - Servlet and JSP engine
tomcat6-admin - Servlet and JSP engine -- admin web applications
tomcat6-common - Servlet and JSP engine -- common files
tomcat6-docs - Servlet and JSP engine -- example web applications
tomcat6-examples - Servlet and JSP engine -- example web applications
tomcat6-user - Servlet and JSP engine -- tools to create user instances
```

从查询结果中，用户可发现软件包 tomcat 5.5 和 tomcat 6。这几个包就是目标的软件包。假如用户打算安装 tomcat 6，在安装的时候，用户只要在使用 apt-get install 时，参数输入为 tomcat6 即可。

例 11.14：查看软件包 tomcat 6 的详细信息

```
sudo apt-cache show tomcat6
```

例 11.15：查看软件包 tomcat 6 与其 WB 软件包之间的依赖关系

```
sudo apt-cache showpkg tomcat6
```

4. 安装软件包

软件安装操作是用户经常使用的操作，使用 apt-get 进行软件包的安装也是比较频繁的。apt-get 很好地解决了 Ubuntu 发布版本中的软件包的依赖关系问题。只要用户安装的软件包是由 Ubuntu 发行版本提供的，就一定可以找到软件包相应的依赖关系包，一定可以很顺利地安装。对 Ubuntu 中一些测试的软件包，就不一定可以找到相应的依赖关系包。

使用 apt-get 安装软件包，方便快捷。安装软件的时候，通过上述的命令查找所需的软件包，若 Ubuntu 软件源中有相应的软件包，就可以使用 apt-get 来安装此软件包。

apt-get 安装软件，使用 install 命令，并配合其他的选项，完成软件包的安装任务。安装

软件包，我们使用的参数是软件包的名称，而不是像 Deb 文档的文件名。这些包都是来自 /etc/apt/sources.list 里边配置的软件源。

例 11.16：安装 rpm 软件包

使用 apt-get install 命令安装 rpm 软件包，执行命令如下所示：

```
apt-get install rpm
```

本节以安装 tomcat 6 软件包为例，说明 apt-get 的软件包安装操作。

例 11.17：只下载软件包而不解压和安装软件包的操作

只下载 tomcat6 软件包而不解压和安装软件包，可执行命令如下所示：

```
dranyu@dranyu-desktop:/etc/apt$ sudo apt-get -d install tomcat6
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
下列软件包是自动安装的并且现在不再被使用了：
  kdelibs4c2a linux-headers-2.6.28-11 kdelibs-data liblualib50 skim
  linux-headers-2.6.28-11-generic libavahi-qt3-1 libskim0 libqt3-mt liblua50 libaudio2
使用'apt-get autoremove'来删除它们
将会安装下列额外的软件包：
  ...
建议安装的软件包：
  ...
下列【新】软件包将被安装：
  ...
共升级了 0 个软件包，新安装了 25 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 59.2MB 的软件包。
解压缩后会消耗掉 151MB 的额外空间。
您希望继续执行吗？[Y/n]
获取: 1 http://cn.archive.ubuntu.com jaunty/main openjdk-6-jre-lib 6b14-1.4.1-0ubuntu7
[4770kB]
获取: 2 http://cn.archive.ubuntu.com jaunty/main ca-certificates-java 20081028 [98.6kB]
...
获取: 25 http://cn.archive.ubuntu.com jaunty-updates/main tomcat6 6.0.18-0ubuntu6.1
[25.0kB]
下载 59.2MB，耗时 47min 24s (20.8kB/s)
下载完毕，目前是“仅下载”模式
```

我们下载的软件包会保存在 /var/cache/archives 目录下，软件的名字为 tomcat 6，我们下载下来的文档名称为 tomcat6_6.0.18-0ubuntu6.1_all.deb。系统同时会自动下载其他有关联的软件包。安装的时候，我们只使用 tomcat 6 软件包名作为 install 的参数。

例 11.18：安装 tomcat 6 软件包

安装时，使用 sudo apt-get install tomcat6 命令。在安装开始之前，用户需要输入自己的账号和密码，之后系统会列出一系列的信息，用户可以根据信息来决定是否需要安装此软件包，提示是否继续，是按 Y，然后按回车键就完成了，剩下的工作都由系统自动完成。执行过程的显示如下：

```
dranyu@dranyu-desktop:/etc/apt$ sudo apt-get install tomcat6
[sudo] password for dranyu:
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
下列软件包是自动安装的并且现在不再被使用了:
```

```

kdelibs4c2a linux-headers-2.6.28-11 kdelibs-data liblualib50 skim
.....
tzdata-java
建议安装的软件包:
.....tomcat6-examples
下列【新】软件包将被安装:
..... tzdata-java
共升级了 0 个软件包, 新安装了 25 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/59.2MB 的软件包。
解压缩后会消耗掉 151MB 的额外空间。
您希望继续执行吗? [Y/n]y
选中了曾被取消选择的软件包 openjdk-6-jre-lib。
(正在读取数据库 ... 系统当前总共安装有 161189 个文件和目录。)
正在解压缩 openjdk-6-jre-lib (从 .../openjdk-6-jre-lib_6b14-1.4.1-0ubuntu7_all.deb) ...
.....
选中了曾被取消选择的软件包 openjdk-6-jre-headless。
正在解压缩 openjdk-6-jre-headless (从 .../openjdk-6-jre-headless_6b14-1.4.1-0ubuntu7_i386.deb) ...
选中了曾被取消选择的软件包 icedtea-6-jre-cacao。
正在解压缩 icedtea-6-jre-cacao (从 .../icedtea-6-jre-cacao_6b14-1.4.1-0ubuntu7_i386.deb) ...
选中了曾被取消选择的软件包 libcommons-daemon-java。
正在解压缩 libcommons-daemon-java
.....
* Starting Tomcat servlet engine tomcat6 [ OK ]
正在处理用于 libc6 的触发器...
ldconfig deferred processing now taking place

```

由于我们之前下载了相关的软件包, 系统会自动识别软件包是否已下载。若软件包已下载, 系统就会直接从/var/cache/apt/archives 目录提取文件并对软件包进行解压, 同时安装配置软件包, 完成软件包的安装。安装了 tomcat 6 之后, 我们在 Firefox 浏览器的地址栏上输入如下链接: <http://localhost:8080>, 若页面上显示 “It works!”, 就表示我们安装 tomcat 6 成功了。或是使用 dpkg -l tomcat 6 命令查看 tomcat 6 软件包的情况, 所查看的状态标志应为 ii。

补充说明

我们不仅可以安装一个软件包, 也可以在一条命令行中同时安装多个软件包, 中间用空格隔开即可。如 sudo apt-get install package1 package2 package3。安装的软件包会默认存放在/var/cache/apt/archives 目录下, 以便以后重新安装。如果已安装的软件包损坏了, 用户可通过--reinstall 选项来重新安装。如 sudo apt-get --reinstall install package_name。

5. 卸装软件包

同样, 使用 apt-get 删 除软件包也有两种方式: 一种是删除软件包, 保留配置文件信息; 另一种是彻底删除, 名为清除, 删除软件包和配置文件信息。上一小节中, 我们讲述了使用 apt-get 安装 tomcat 6 软件包, 在这一小节中, 我们就来讲述卸装 tomcat 6 软件包。

例 11.19: 删 除 tomcat 6 软件包

若卸装的时候, 需要保留 tomcat 6 软件包的配置文件, 用户打算以后还需要使用 tomcat6 作为 Web 服务器, 我们就使用 apt-get remove 命令来对软件包进行卸装。卸装的命令很简单,

效率也很高，输入如下命令：

```
dranyu@dranyu-desktop:/etc/apt$ sudo apt-get remove tomcat6
```

提示输入密码：

```
[sudo] password for dranyu:
```

输入密码之后，显示软件包的卸装情况。

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

下列软件包是自动安装的并且现在不再被使用了：

```
.....libaudio2
```

使用'apt-get autoremove'来删除它们

下列软件包将被【卸载】：

```
tomcat6
```

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 258KB 的空间。

这里 apt 提示，我们只卸装了 tomcat 6 这个软件包，可是还有其他相关的软件包没有被卸装，而且这些软件包不再被其他程序使用。这样使用 apt-get autoremove 命令就可以删除这些冗余的软件包。使用 dpkg -l tomcat 6 命令查看 tomcat 6 软件包的情况，标志为 rc，表示 tomcat6 软件包已从系统上被删除。

使用 sudo apt-get --reinstall install tomcat 6 命令重新安装 tomcat 6 软件包。

例 11.20：清除 tomcat 6 软件包

接着我们来一次完全卸装，清除 tomcat 6 软件包，执行过程和结果如下：

```
dranyu@dranyu-desktop:/etc/apt$ sudo apt-get remove tomcat6 --purge
```

下列软件包是自动安装的并且现在不再被使用了：

```
.....libaudio2
```

使用'apt-get autoremove'来删除它们

下列软件包将被【卸载】：

```
tomcat6*
```

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 258kB 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 162756 个文件和目录。)

正在删除 tomcat6 ...

[OK]

* Stopping Tomcat servlet engine tomcat6

正在清除 tomcat6 的配置文件 ...

正在删除用户 'tomcat6'...

警告：组 'tomcat6' 没有多的成员了。

完成。

dpkg - 警告：当删除“tomcat6”，目录“/var/lib/tomcat6/webapps”非空，因而不会删除该目录。

dpkg - 警告：当删除“tomcat6”，目录“/var/lib/tomcat6”非空，因而不会删除该目录。

从显示的信息中我们可以看到系统确实把 tomcat 6 软件包清除了。我们还可以使用 dpkg -l tomcat6 命令来确定 tomcat 6 软件包是否处于清除状态，可以查看到状态为 Pn，证明配置也删除了。

6. 其他命令

命令 sudo apt-get upgrade 用于更新已安装的包，对已安装的软件包进行升级操作。

命令 sudo apt-get dist-upgrade 用于升级系统。

命令 sudo apt-get dselect-upgrade 使用 dselect 的升级系统，升级已安装的软件包。



命令 `apt-cache depends package` 用于了解软件包所依赖的软件包。

命令 `apt-cache rdepends package` 用于查看该软件包被哪些包依赖。

命令 `sudo apt-get build-dep package` 用于安装相关的编译环境。

命令 `apt-get source package` 用于下载该软件包的源代码。

命令 `sudo apt-get clean` 和 `sudo apt-get autoclean` 用于清理无用的软件包。

命令 `sudo apt-get check` 用于检查是否有损坏的依赖。

11.2.3 Aptitude 软件包管理工具

Aptitude 是一个菜单驱动，基于文本的 Advanced Packaging Tool (APT) 系统前端。软件包管理的许多常用功能，如安装、卸载和升级，可以在 Aptitude 中单击菜单执行命令，它通常是小写字母。Aptitude 最适用于非图形的终端环境，确保命令关键字的适当功能。

Aptitude 可以在终端执行如下命令来完成管理软件包的任务：

```
aptitude update 更新可用的包列表。  
aptitude upgrade 升级可用的包。  
aptitude dist-upgrade 将系统升级到新的发行版。  
aptitude install pkgname 安装包。  
aptitude remove pkgname 删除包。  
aptitude purge pkgname 删除包及其配置文件。  
aptitude search string 搜索包。  
aptitude show pkgname 显示包的详细信息。  
aptitude clean 删除下载的包文件。  
aptitude autoclean 仅删除过期的包文件。
```

同时，用户也可以通过在终端如数 `sudo aptitude` 启动 Aptitude 的字符界面菜单操作功能，让用户更方便管理软件包。在 Aptitude 程序下所有的操作都是通过单击菜单完成。终端输入 `sudo aptitude`，之后输入用户密码，就可以登录如图 11-1 所示的操作界面。

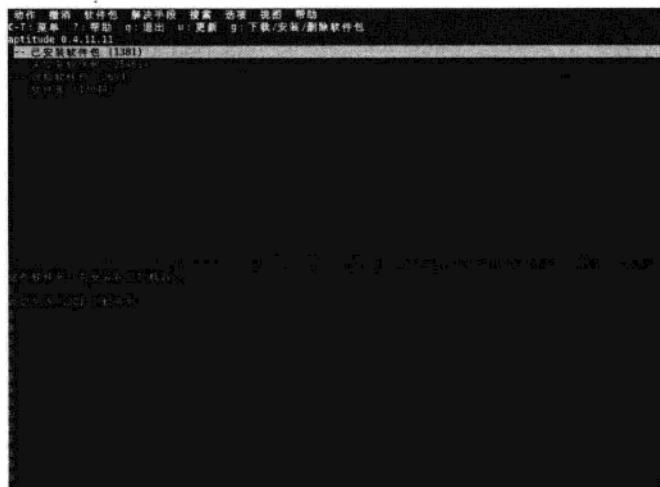


图 11-1 Aptitude 操作界面

1. 配置软件源

Aptitude 与 apt-get 使用的是统一的软件源配置文件。因此，不再重述 Aptitude 中软件源的配置。

2. 更新软件源

配置好软件源之后，通过单击 Aptitude 操作界面的菜单完成操作。选择“动作”|“更新软件包列表”菜单命令，系统就会自动执行软件包列表的更新。实际上，系统在后台执行了 Aptitude update 命令。

3. 查找软件包

查找软件包在 Aptitude 下也成为很简单的事情。Aptitude 提供了多种软件包查找方式，其中包括如下两种常用方式：

- 选择“搜索”|“查找”菜单命令或是按下“/”键，表示向后查找，在提示框中输入所要安装的软件包的名称或关键字。执行之后中间分类框显示查询信息。如第一次没有查询到所需的软件包，重复上述操作，继续查询。
- 选择“搜索”|“限定显示”菜单命令，在弹出的对话框中输入软件包的名称或关键字，执行操作。这里使用的限定显示，输入软件包名称或是关键字之后，中间的包分类，就会只显示有关与输入字符相匹配的软件包分类，其他分类就不显示了。可以看到执行查找 tomcat 6 软件包操作之后，结果如图 11-2 所示。



图 11-2 查找 tomcat 6 软件包

可以看到数据库中将软件包 tomcat 6 的状态标识为 P，表示系统已安装过 tomcat 6，但后来被卸装了，并且被清除。

4. 安装软件包

仍然以安装 tomcat 6 软件包为例，讲述 Aptitude 安装软件。通过上一步骤，我们查找到 tomcat 6 软件包。双击 tomcat 6 软件包名，Aptitude 就会在另一个窗口中显示 tomcat 6 的详细信息如图 11-3 所示。

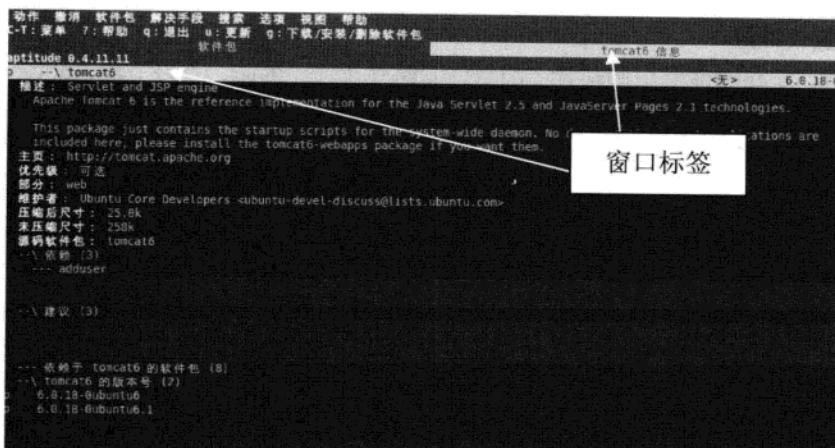


图 11-3 tomcat 6 软件包信息

第一条白色的横条表示当前视图的标签，而“tomcat6 信息”为此视图的名称。第二条白色的横条指明查看的 tomcat 6 软件包，其中左边的 P 表示此软件包已被清除。红色区域表示软件包的依赖关系。

选中第一行之后，选择“软件包”|“安装”菜单命令，对 tomcat 6 软件包进行安装软件包的标记。标记之后，软件包就会发生变化，并且软件包的状态信息更改为 pi，操作结果如图 11-4 所示。

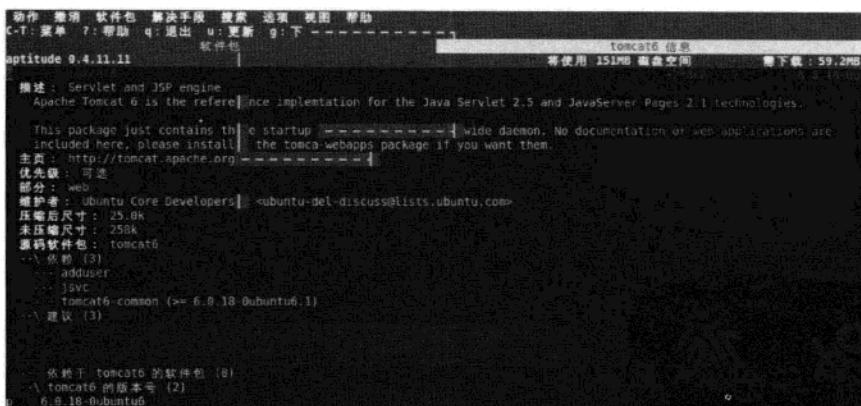


图 11-4 标记安装 tomcat 6 软件包

此时 Aptitude 只是对所要执行操作的软件包作了标记，Aptitude 并没有安装 tomcat 6 软件包。唯有在选择“动作”|“安装/卸载软件包”菜单命令或是按下“G”键之后，安装的操作才会开始执行。第一次按下“G”键时 Aptitude 会将要安装或卸载的软件包列表显示在屏幕上，第二次按下“G”键时 Aptitude 才会执行安装或卸载的操作。执行安装软件包的过程中，Aptitude 会把信息显示在终端上。软件包安装完毕之后，需要按下回车键才会回到 Aptitude 的操作界面。这种安装的安装信息显示结果与使用 apt-get install 的命令安装软件包的显示结

果相同。Tomcat 6 软件包安装完毕，可以在“已安装软件包”分类中找到 tomcat 6 软件包的记录。

5. 卸载软件包

Aptitude 提供了删除软件包的两个操作，一个是删除，一个是清除。在 Aptitude 操作界面的菜单栏上选择“软件包”菜单命令，弹出菜单列表如图 11-5 所示。

选择“删除”菜单命令，对软件包进行删除操作的标记，标记卸装软件包之后，对话框信息显示如图 11-6 所示。

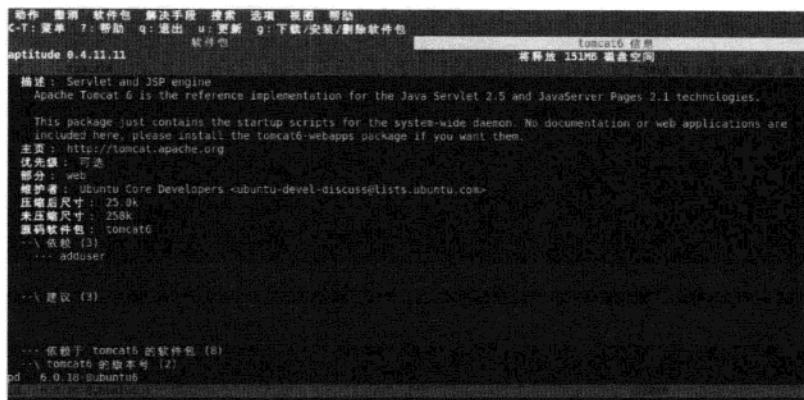
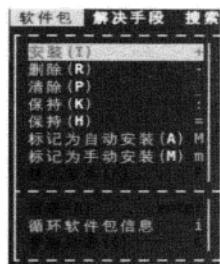


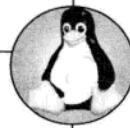
图 11-5 “软件包”菜单

图 11-6 标记删除软件包

可以看到顶部首行的行头的两个字符 id 表示软件包的状态。i 表示已安装，b 表示即将删除。执行删除操作的方法为：选择“动作”|“安装/卸载软件包”菜单命令或按下“G”键执行操作。Aptitude 自动转到终端下运行，并在终端上显示软件包安装过程的信息。删除软件包之后，软件包的状态标志为 c，表示已删除软件包，可是还保留了配置。

若要清除软件包，那操作如同删除操作，只是在选择菜单时选择“清除”菜单命令。清除完毕之后，软件包的状态标识为 p，表示软件包已清除。清除软件包操作在终端显示结果如下：

```
dranyu@dranyu-desktop:/etc/apt$ sudo aptitude
[sudo] password for dranyu:
(正在读取数据库 ... 系统当前总共安装有 141631 个文件和目录。)
正在删除 icedtea-6-jre-cacao ...
正在删除 tomcat6 ...
* Stopping Tomcat servlet engine tomcat6
[ OK ]
正在删除 tomcat6-common ...
.....
正在删除 libgcj9-0 ...
rmdir: 删除 “/var/lib/gcj-4.3” 失败: 没有该文件或目录
.....
正在处理用于 doc-base 的触发器...
Processing 5 removed doc-base file(s)...
Registering documents with scrollkeeper...
正在处理用于 man-db 的触发器...
```



```
正在处理用于 libc6 的触发器...
ldconfig deferred processing now taking place
按回车继续。
```

从操作显示出来的结果，我们可以看到 Aptitude 执行清除操作之后，同时也把 tomcat 6 软件包所依赖的软件包删除了。这样用户就不必再多执行一次 autoremove 命令，来清理那些没用的失去关联的软件包。

要退出 Aptitude，只需简单按“Q”键并确认想退出即可。

补充说明

当实际查看软件时列出软件包当前状态，在顶窗软件包列表中显示信息的第一列使用下列关键字来描述软件包状态：

- i: 安装软件包
- c: 软件包没有安装，但在系统中有软件包的残留配置
- p: 从系统彻底删除
- v: 虚拟软件包
- B: 已损坏的软件包
- U: 解压文件，但尚未配置软件包
- C: 半配置 - 配置失败需要修复
- H: 半安装 - 卸载失败需要修复

Aptitude 软件包管理器提供了许多快捷键，这些快捷键能够帮助用户掌握 Aptitude 的操作。启动 Aptitude 之后，按下“Shift+ /”组合键，可查看 Aptitude 的帮助手册。帮助手册详细地列出了 Aptitude 快捷键的含义。

11.2.4 其他软件包管理工具

Ubuntu 还提供了其他的软件包管理工具，如 dselect 和 tasksel 工具。dselect 软件包管理工具与 Aptitude 同样是高级的前端管理工具。dselect 和 tasksel 工具都是使用/var/lib/dpkg 保存软件包的包信息的。这与 apt 和 Aptitude 使用/var/lib/apt/lists/保存包信息有所不同。

dselect 的操作也很简单，与 Aptitude 相当，都为菜单操作界面，操作界面如图 11-7 所示。

按下键盘上的方向键就可以选择不同的选项，之后按下回车键，dselect 就会执行相应的操作，包括选择软件包、安装软件包、配置软件包和删除软件包等。dselect 的使用是循序渐进的，按照步骤一步一步地完成，期间 dselect 会向用户提供仅有 5 屏的用户帮助。

在查找功能中，dselect 向用户询问几个问题，就可以向用户显示用户所需的相关软件包信息，dselect 的智能性比较好。找到定制的软件包之后，用户可以根据提示选择所要进行的操作，如安装、删除、清除和升级。

安装过程中，dselect 自动检测软件包的依赖关系，若有什么问题，如缺少关联包，或是有软件包冲突，dselect 都会给出善意的提示，请求用户做出选择。

“删除”选项，删除不必要的软件包，也就是清除其他无用的软件包。

dselect 在完成指定操作后，会自动退出 dselect 应用程序。



图 11-7 dselect 操作界面

在 Ubuntu 9.04 中默认没有安装 dselect，若用户喜欢使用 dselect 工具，可以使用 sudo apt-get dselect 命令安装 dselect 工具包。

若用户需要安装软件包套装，使用 tasksel 来完成安装是一个不错的选择。在终端输入命令行 sudo tasksel 就可以进入如图 11-8 所示的 tasksel 安装软件包界面。

使用 tasksel 管理软件集，操作简单，工作效率高。上图中通过按下方向键移动红色光标，按下 space 空格键选择或取消选择软件集。软件包第一次安装时，按下 space 空格键，则选项前端就会有一个*号作为标记，按下“确定”按钮，就开始进行安装所选定的软件包。

已安装的软件集都会标上两个*符号。若需要卸载软件集，再把光标移动到相应的软件集上，可以发现两个*标号，按下空格键，就可去掉一个*符号，标识此软件集即将被卸载，单击“确定”按钮，就可以卸载软件集了。tasksel 的卸载为删除操作，保留着软件包的配置。

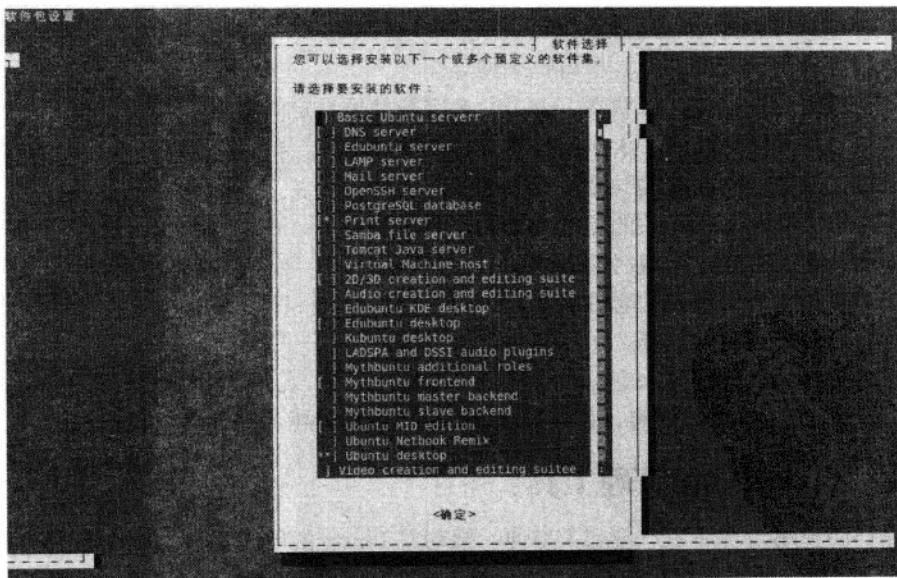


图 11-8 tasksel 工具软件选择

dpkg 是 Debian 系统中最底层的软件包管理工具，APT 机制是 Debian 系统高级的软件包管理工具，dselect 是建立在 dpkg 之上的，而 Aptitude 是以 apt-get 作为基础工具。每个软件



包管理工具都具有各自的特点，而且每种工具并不冲突。`apt-get` 使用执行效率高，在卸载软件包时，并不会删除它所依赖的软件包，而 `Aptitude` 做到了这一点，而且 `Aptitude` 的菜单式操作也大大方便了用户操作。`Aptitude` 很好地保证了软件包的依赖关系，而且维护着系统的清洁，减少系统中软件包的冗余。目前经过多人的测试，觉得 `Aptitude` 要比 `dselect` 好用，也许就因为这个原因 Ubuntu 默认只安装了 `Aptitude` 软件包管理工具。用户可以根据自己的爱好，选择不同的软件包管理工具。

11.3 新立得软件包管理工具



新立得软件包管理器起源于 Debian。在 Ubuntu 中，新立得是 Ubuntu 的包管理工具 APT 的图形化前端，以 GTK+编写，结合了图形界面的简单操作和 `apt-get` 命令行工具的强大功能。用户可以使用新立得安装、删除、配置、升级软件包，对软件包列表进行浏览、排序、搜索以及管理软件仓库或者升级整个系统。用户可以进行一系列操作形成操作队列，并一起执行它们。新立得会提示用户可能的依赖关系（用户所选择的软件包所需的额外软件包），以及用户所选择的软件包与系统中已经安装的软件包的冲突。

新立得软件包管理器的特点如下：

- 提供一个易于安装、移除套件，为系统升级或降级的图形使用者接口的工具。
- 可以将整个系统升级。
- 管理套件来源 (`sources.list`)。
- 可以依名字、简介及其他特点来搜寻套件。
- 可以设定过滤器，以状态、分类、名字或其他自定的特点来选取套件。
- 可以依名字、状态、大小或版本来排列套件。
- 可以浏览所有在线的与套件相关的说明文档。
- 可以下载与套件相关的更近变更的资料。
- 将特定套件锁定于现时的版本。
- 可以指定安装特定的套件版本。
- 可以复原或重做选择。
- 内置终端机模拟器。
- 只是 Debian 系统所有，可以经由 `debconf` 系统对相关套件做出设置。

11.3.1 启动新立得管理工具

新立得的使用给普通用户带来了方便，推动着 Linux 的发展。启动新立得图形界面软件包管理工具，可以通过在终端执行 `sudo synaptic` 命令，也可以通过选择 Ubuntu 顶部面板上的“系统” | “系统管理” | “新立得软件包管理器”菜单命令，两种方式来启动。启动之后，显示在用户面前的操作界面如图 11-9 所示。

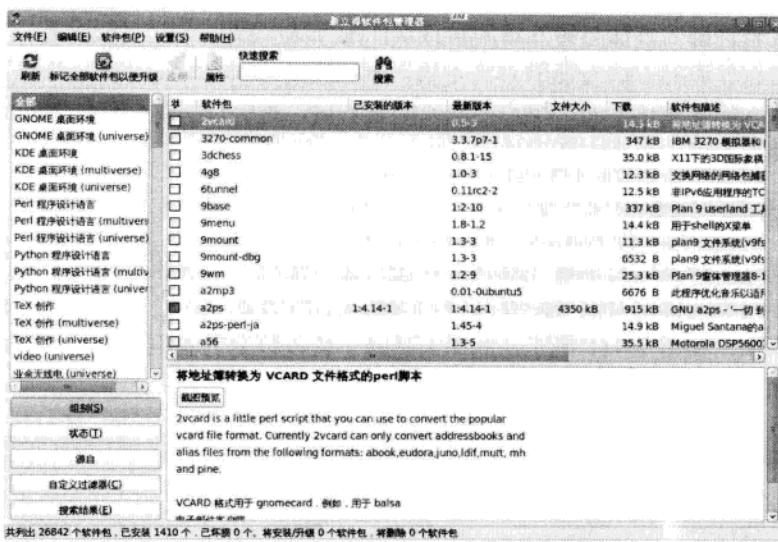


图 11-9 新立得软件包管理器主界面

补充说明

由于新立得软件包管理器所做的操作都涉及系统配置的修改，所以它的运行需要进行用户认证，以获取系统管理员的操作权限。在提示输入用户名时，输入用户的密码，通过系统认证之后，新立得软件包管理器就可执行软件包的安装、修改和删除操作。

11.3.2 设置软件源

由于新立得是基于 apt-get 的，所以它使用与 apt-get 同样的数据源文件 /etc/apt/sources.list。在 Ubuntu 中，系统同时提供了一个图形的配置软件源的工具。通过在新立得的主界面选择“设置” | “软件库”菜单命令，同样弹出如图 11-10 所示的设置软件源窗口。



图 11-10 设置软件源窗口



此处设置软件源的方法与第 6 章的设置软件源方法一致。

上述配置软件源的方法与使用 vi 文本编辑器编辑/etc/apt/sources.list 文件的方法效果是一样的。使用这一配置图形界面，用户在修改配置之后，系统自动执行软件源的更新动作，无须用户在终端再次输入命令更新软件源。

11.3.3 查找软件包

新立得提供了智能的查询软件包功能。在快捷工具栏上的快速搜索栏中输入关键字，这样新立得就会根据关键字对系统软件源中的软件包进行过滤并将过滤结果实时显示到右边的软件包列表框中，如图 11-11 所示，此图为选择了软件包浏览器中“全部”菜单命令分类的搜索结果界面。

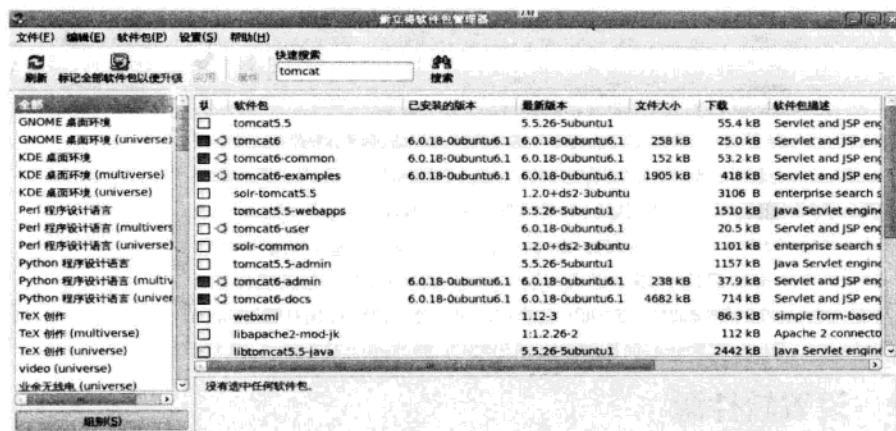


图 11-11 快速搜索

当快速搜索栏中有关键字的时候，用户单击软件包浏览器中的软件包分类，新立得也会将过滤操作实施到该分类中。若用户不希望软件包被过滤，必须把快速搜索栏清空。

用户单击快速工具栏上的“搜索”按钮，弹出查找界面如图 11-12 所示。

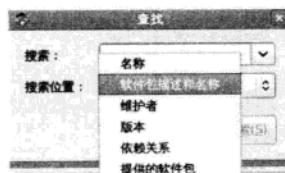


图 11-12 查找界面

在对话框的搜索栏中输入软件包的关键字，选择相应的搜索位置，输入 tomcat 6，选择软件包描述和名称作为搜索位置，之后单击“确定”按钮或按下回车键，对关键字进行搜索。搜索结果将在右边的软件包列表框中显示“tomcat 6”的分类，显示结果如图 11-13 所示。

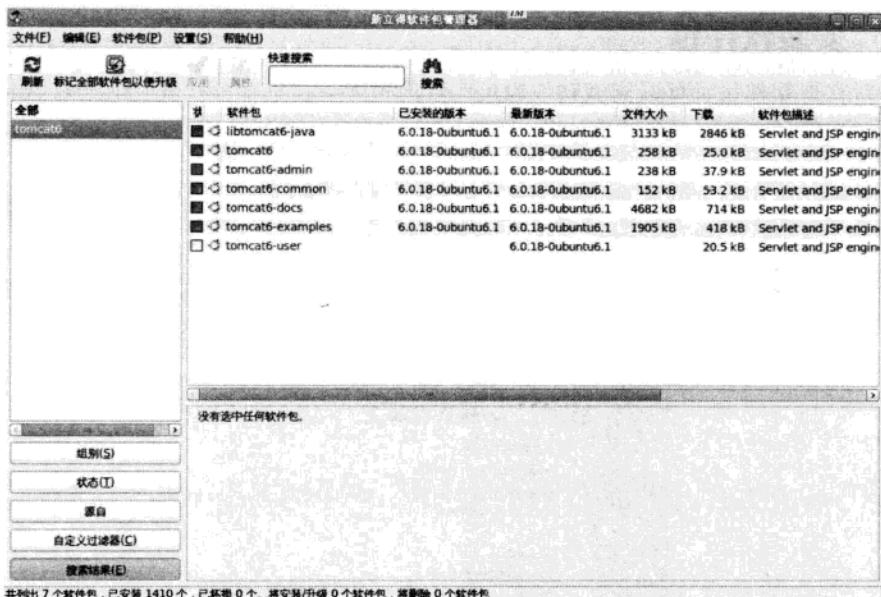


图 11-13 搜索结果

新立得罗列出匹配的软件包，我们可以通过单击软件包记录查看软件包的属性，具体的操作方法为：选中第二行记录——tomcat 6 软件包，然后单击快捷工具栏上的“属性”按钮。tomcat6 软件包的详细信息显示在 tomcat 6 属性窗口中如图 11-14 所示。

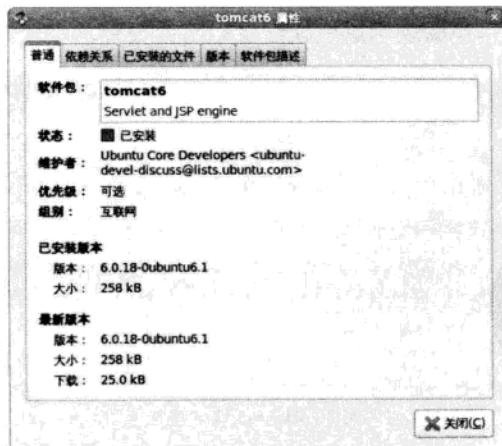


图 11-14 tomcat 6 软件包的属性

在 tomcat 6 属性窗口的“普通”选项卡上，显示了软件包的名称，软件包的状态信息等；在“依赖关系”选项卡中主要说明了此软件包的依赖关系；在“已安装的文件”选项卡中，列出了所有相关文件的安装路径；在“版本”选项卡中罗列出在软件源中可用的软件包版本号。



11.3.4 安装软件包

使用新立得软件包管理器安装软件包是比较简单的。在本小节图 11-17 中，我们可以看到系统已经安装了 tomcat 6 软件包。为了说明软件包的安装问题，我们选择安装 apache 作为例子。

通过前面搜索软件包的步骤查找到 apache 软件包，接着为 apache 软件包作标记，最后应用所作的标记软件包的安装操作，执行安装软件包。

查找到 apache 软件包，双击软件包记录，弹出如图 11-15 所示的提示对话框。

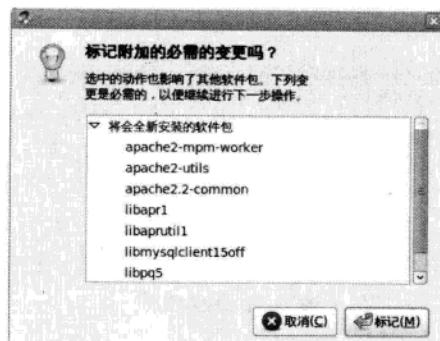


图 11-15 提示标记对话框

新立得管理器会自动检测软件包的依赖关系，系统会自动下载和安装那些软件包作为附加必须安装的软件包，单击“标记”按钮，新立得就会对相关联的软件包作标记，作了标记的软件包记录会变为绿色，也即表示这些软件包即将被安装，显示效果如图 11-16 所示。

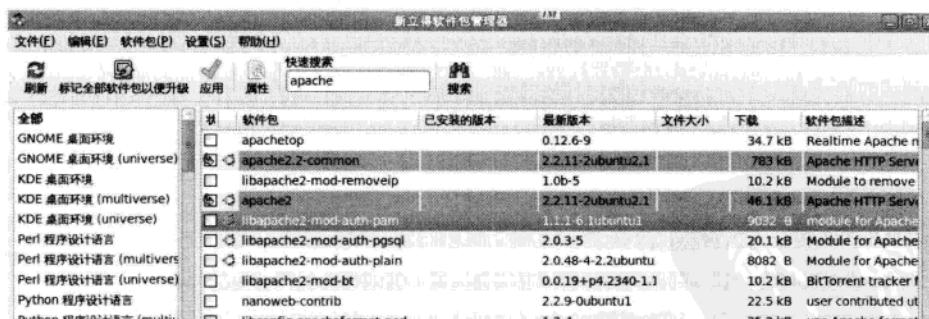


图 11-16 标记软件包

所需安装的软件包都被作了标记，现在开始执行安装操作。单击工作栏上的“应用”按钮执行安装操作，剩下的工作都由新立得管理工具完成。

新立得检测安装的相关软件包，从软件源中下载软件包，然后解压软件包，配置系统环境，最后向系统安装程序。软件包安装完毕之后新立得弹出了一个任务完成对话框，如图 11-17 所示。

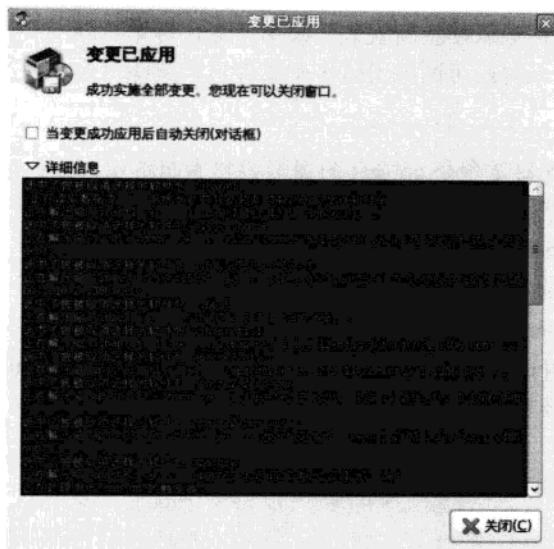


图 11-17 安装软件包完毕

可以在详细信息中看到，新立得的安装软件包显示的结果与使用 `apt-get` 安装软件包时显示的结果是一致的。单击“关闭”按钮完成了 `apache` 软件包的安装。在 `firefox` 的地址栏上输入：`http://localhost`，按下回车键，可以看到页面上显示“`It works`”，表示 `apache` 服务器已经成功安装，并正在运行中。

11.3.5 卸载软件包

软件包的卸载分为删除软件包和彻底删除软件包（清除软件包）。安装了 `apache` 之后，现在我们以卸载它作为例子讲述如何使用新立得软件包管理器卸装软件包，具体的操作步骤如下所示：

(1) 通过新立得的查找功能，找到 `apache` 软件包的记录。查看它是否已安装，在新立得软件包管理工具中，已安装的软件包，都会在软件包记录的第一列显示为绿色的小方块。此处 `apache` 已经安装，必能看到 `apache` 软件包记录前已被标识为已安装。

(2) 用鼠标右键单击已安装的 `apache` 软件包记录，弹出如图 11-18 所示的菜单。对于删除操作，我们可以选择“标记以便删除”菜单命令，若我们要彻底删除，那么我们可以选择“标记以便彻底删除”选项。

新立得的软件包卸载就是如此简单，作个标记，之后就单击快捷工具栏上的“应用”按钮，执行用户要求新立得所进行的操作。执行删除操作，实际上是在终端上执行了 `apt-get remove apache2`。而执行彻底删除则是执行了 `apt-get remove apache2 --purge`。新立得执行删除和彻底删除操作时，都只是卸载选定的软件包，并未同时删除此软件包所依赖的其他软件包。



图 11-18 弹出的快捷菜单



同样用户可选择“标记以便重新安装”菜单命令，标记重新安装软件包。除了可以安装和卸载单个软件包，新立得还可同时安装多个软件包，删除多个软件包。

补充说明

新立得软件包管理器是命令 apt-get 的图形操作界面前端，在使用新立得软件包管理器的时候，在终端是不可以使用命令 apt-get 的。如下所示：

```
dranyu@dranyu-desktop:/$ sudo apt-get install vim
```

```
[sudo] password for dranyu:
```

```
E: 无法获得锁 /var/lib/dpkg/lock – open ( 11 资源临时不可用 )
```

```
E: Unable to lock the administration directory ( /var/lib/dpkg/ ), is another process using it?
```

此时，新立得软件包管理器正在运行中。



11.4 小结

Ubuntu 提供了多种多样的软件包管理工具，用户可以在具体的情况下使用其中之一。dpkg 命令行执行方便，作为基础的管理工具，功能强大。其他的管理工具，都是直接或间接使用了它。apt-get 在最底层也是调用了 dpkg 命令。各个软件包管理器之间并不相互排挤，而且一起协调工作。dselect 和 tasksel 管理工具直接使用了 dpkg 命令，并使用了 dpkg 的数据库包的信息。

apt-get、Aptitude 和 Synaptic 新立得软件包管理工具都是 APT 机制的，都使用了 /etc/apt/sources 作为软件源配置。Aptitude 解决了 apt-get 执行删除时没有删除相应的依赖软件包，Aptitude 在维护系统的清洁上是占优势的。Synaptic 软件包管理工具是一个图形界面的前端 APT 管理工具，apt-get 很多功能都集成到这个软件中，用户只要单击图形菜单就可以完成所要执行的功能，并不需要有过多的命令行知识。



11.5 习题

- 在 Ubuntu 中软件包有多少种类型？Deb 软件包的格式中的每个字段的含义是什么？
- Deb 软件包之间的依赖关系有多少种？各表示什么含义？
- 什么是虚拟软件包？它的作用是什么？
- Ubuntu 中有几种软件包管理工具？它们在运行的时候都调用了哪些命令？
- 使用 Gedit 应用软件打开/etc/apt/sources.list 文件，查看此文件内容的编写格式。
- 使用命令 sudo aptitude 启动 Aptitude 应用程序。
- 熟悉 apt-get 命令的安装软件操作和卸载操作。
- 启动新立得软件包管理器，搜索 vim 软件包，查看此软件包的属性，然后安装此软件包。

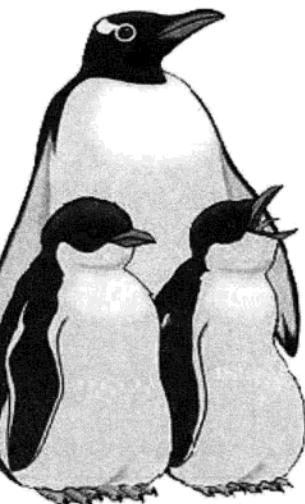
Linux

第12章

常用应用软件

Ubuntu Linux 中提供给用户多种多样的应用软件，如编辑软件、办公软件、多媒体软件、娱乐软件、网络浏览器和下载工具等。这些软件都属于上层的扩展软件，都是基于 Ubuntu 的底层软件，且经过 Ubuntu 社区测试，并具有良好的性能。

Ubuntu 下的常用软件基于 GNOME 桌面而开发，在这一章主要讲述 Ubuntu 下常用应用软件的使用，使读者阅读完这一章之后，能够在 Ubuntu 中顺利地进行日常生活、工作和娱乐活动。本章主要介绍的常用应用软件包括文件编辑工具 Gedit、办公软件 OpenOffice.org 套件、图片管理与处理软件、多媒体播放软件、即时通信软件、下载工具软件和浏览器软件等。





12.1 文本编辑工具 Gedit

在 Windows 系统下，用得比较多的文本编辑器，莫过于 notepad 和记事本，而在 Ubuntu Linux 下，用得比较多的文本编辑器莫过于 Gedit。

Gedit 文本编辑器的使用很频繁，它能打开 txt 文本和 html 文件等。用户在使用计算机的时候一定会与文本编辑器打交道。它是 GNOME 桌面环境下兼容 UTF-8 的文本编辑器，使用 GTK+ 编写而成，因此它十分简单易用，有良好的语法高亮，对中文支持很好，支持包括 gb2312、gbk 在内的多种字符编码，并且它是一个自由软件。Ubuntu9.04 默认安装了 Gedit 文本编辑器。

Gedit 文本编辑器的功能要比 Windows 下的文本编辑器强大。Gedit 支持语法高亮和标签编辑多个文件的功能。利用 GNOME VFS 库，它还可以编辑远程文件。它支持完整的恢复与重做系统，以及查找与替换功能，还支持包括多语言拼写检查和一个灵活的插件系统，可以动态地添加新特性，例如与外部程序的整合。Gedit 还包括一些小特性，例如行号显示，括号匹配，文本自动换行，当前行高亮，以及自动文件备份。

12.1.1 启动 Gedit

在 Ubuntu 系统下，使用 Gedit 编辑文件时，可通过如下方式启动 Gedit 应用程序：

- (1) 选择主界面的“应用程序”|“附件”|“文本编辑器”菜单命令。
- (2) 在终端输入如下命令：

```
Gedit  
或 Gedit &
```

以上述方式启动 Gedit 之后，Gedit 应用程序自动创建一个名为“未保存文档 1”的文本文件，其主界面如图 12-1 所示。

用户可看到 Gedit 的主窗口由 6 行组成：顶行为标题栏；第二行为菜单栏；第三行为快捷工具栏；第四行为文件标签；第五行为文本显示框，最后一行为状态栏。

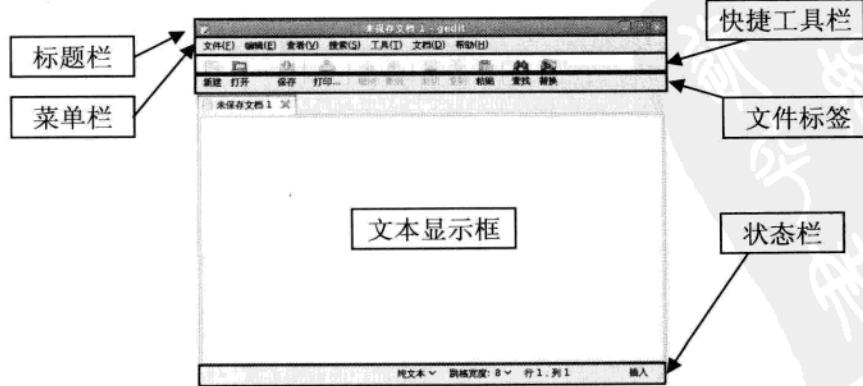


图 12-1 Gedit 主界面

- 标题栏

标题栏中显示文件名、文件的路径和 Gedit 字符串。

- 菜单栏

菜单栏放置 Gedit 的功能菜单。Gedit 处理文件时所用的所有功能都可以在菜单栏中找到相应的菜单。在帮助菜单上可以找到有关 Gedit 文本编辑器的相关信息和帮助文件。

- 快捷工具栏

快捷工具栏放置了常用的文本编辑工具，包括撤销、重做、剪切、复制、粘贴、查找和替换等。单击快速工具栏上的快捷按钮，可快速执行相应的操作。

- 文件标签

文件标签栏上会显示 Gedit 所打开的多个文件的名称。其上的每个文件标签都与已打开的文件相对应。文件标签可以被任意拖动，使用户可把使用频繁的文档放置在比较方便操作的位置。单击文件标签可切换到与之相应的文件。

- 文本编辑框

这一区域就是文本的编辑区域，是文本操作的平台，是文本显示的区域。若使用查找功能，匹配的文字会在此框中高亮显示。在文本编辑框中，输入和编辑程序代码时，代码会高亮显示。

- 状态栏

在状态栏中用户可以设置文本高亮显示的方式。单击“纯文本”字样右边的箭头，在下拉菜单中显示多种文件的格式名称，如 C++，html。选中不同的文件格式时文本编辑框中的文本就会以不同的显示形式显示，例如选择了 C++文件格式，文本会以 C++的语法高亮显示。

状态栏上还显示了光标所在的行号和列号。如图 12-1 所示的“插入”字样，表示此时文本的编辑模式为插入状态。若按下“Insert”键，此字样会更改为“覆盖”，表示此时文本的编辑模式为覆盖模式，输入的字符会覆盖光标后面的字符。再次按下“Insert”键，编辑模式再次切换为插入模式。

12.1.2 处理文件

处理文件表示 Gedit 如何创建文件、打开文件、保存文件、关闭文件、打印文件。

1. 创建文件

单击主界面上的“文件”|“新建”按钮，或是单击快捷工具栏上的“新建”按钮，或按下“Ctrl+N”快捷键，创建一个新的文本文件。打开 Gedit 的同时，Gedit 默认创建一个题为“未保存文档 1”的新文档。Gedit 应用程序就会自动为新建的文本文件命名，一般命名为“未保存文档 n”，n 为一个数字。

用户通过终端，进入工作目录，执行 Gedit newfile.*命令，也可以创建一个名为 newfile.*的新文本。其中*表示文件的后缀。

2. 打开文件

Gedit 所打开的文件可分为本地磁盘上的文件和网络上的文件。以下介绍几种常见的打开文件的方式：

(1) 双击本地磁盘的文件就可以打开文件。



(2) 在 Gedit 应用程序的菜单栏上选择“文件”|“打开”菜单命令，弹出“打开文件”对话框，选择所要打开的文件（此处为 good1 文件），如图 12-2 所示。选择指定的文件之后，单击“打开”按钮，就可打开文件。



图 12-2 浏览所要打开的文件

使用这种方法还可同时打开多个文件，方法是：在“打开文件”对话框中，结合 Shift 或 Ctrl 键在文件列表中选择多个文件，然后单击“打开”按钮。

(3) Gedit 除了可打开本地的文件，还可打开网络上的文件。在 Gedit 应用程序的菜单栏上选择“文件”|“打开位置”菜单命令，弹出“打开位置”对话框，在“输入您想打开文件的位置 (URI) (L) :”的文本输入栏上输入 <http://www.163.com> 字符串，如图 12-3 所示。然后单击“打开”按钮，就可在 Gedit 应用程序中打开 163 网页首页的源代码。



图 12-3 打开网络中的文件

补充说明

Gedit 应用程序的可运行文件为 /usr/bin/Gedit，用户可在终端的命令行提示符中启动 Gedit 应用程序，并打开文件。使用命令 Gedit 打开各种形式文件的例子如下所示：

本地文件的打开：Gedit index.html

网络文件的打开：Gedit <http://www.163.com>

多文件的打开：Gedit file1 file2 file3

3. 保存文件

当编辑文本完毕之后，需要保存对文件的修改。单击快捷工具栏上的“保存”按钮，或选择“文件”|“保存”菜单命令，或按下“Ctrl+S”快捷键，保存文件。若文件是第一次被保存，那么 Gedit 应用程序会弹出“另存为...”对话框，在“名称”的文本输入栏中输入 newfile，在“浏览其他文件夹”选项框中选择 doc 文件夹，如图 12-4 所示。

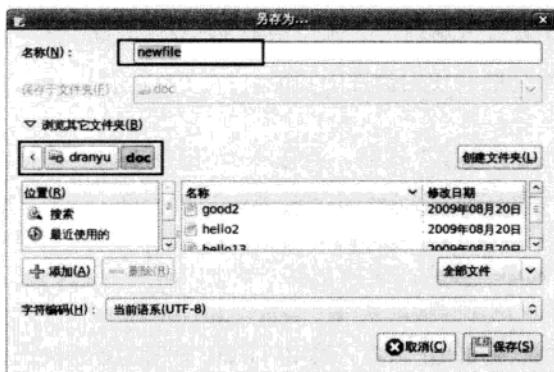


图 12-4 保存文件

4. 关闭文件

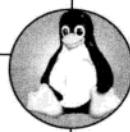
单击文件标签栏上的关闭按钮 \square ，就可以关闭文件；或按下“Ctrl+W”组合键关闭文件。若在终端使用 Gedit 而没有使用&参数，可按下“Ctrl+C”组合键中断 Gedit 的运行，退出 Gedit 应用程序。

5. 打印文件

Gedit 提供了文件打印的功能。选择“文件”|“打印预览”菜单命令，查看文件打印的效果。选择“文件”|“打印”菜单命令，弹出“打印”对话框，如图 12-5 所示。



图 12-5 打印



此处为将文件从打印机中打印出来。若在打印机列表中选择“打印到文件”选项，可将文件输出为.pdf 格式的文件。

12.1.3 处理文本

Gedit 的文本处理功能强大。除了具备基本的文本编辑器的功能（文本的插入、删除、复制、剪切和粘贴）外，还包括其他的高亮突出功能和插入用户与插入时间等插件附加功能，在搜索和替换功能上具有自己的特色，会以鲜明的颜色区分匹配的字符串。此处主要讲述 Gedit 的搜索功能、替换功能和使用插件的功能。

1. 搜索

单击快捷工具栏上的“查找”按钮，或按下“Ctrl+F”组合键，弹出“查找”对话框，如图 12-6 所示。

在搜索栏中输入关键字之后，底部的“查找”按钮变为可用。单击“查找”按钮执行查找操作。若 Gedit 在文本中查找到匹配的文本，就会以鲜明的色彩标识匹配文本。

2. 替换文本

若替换文本中有大量相同的字符串，那么 Gedit 提供的替换功能是一个很好的选择。单击快捷工具栏上的“替换”按钮，或选择“搜索”|“替换”菜单命令，或按下“Ctrl+H”组合键，弹出“替换”对话框如图 12-7 所示。

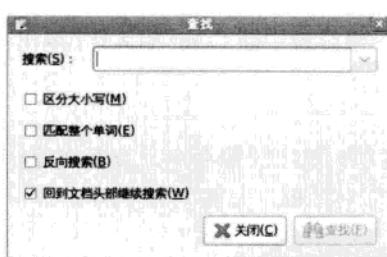


图 12-6 “查找”对话框

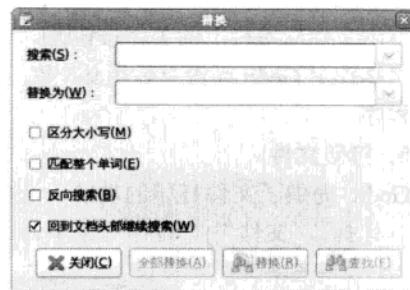


图 12-7 “替换”对话框

在搜索栏中输入需要被替换的字符串，在替换栏中输入替换后的字符串之后，底部的“全部替换”、“替换”和“查找”按钮变为可操作。

单击“全部替换”表示将文本中全部匹配的字符串替换为所需的字符串。单击“替换”表示是替换当前匹配的字符串。而单击“查找”表示从光标处向下或向上继续查找匹配的字符串。

3. 使用插件

在使用插件之前需要将插件添加到 Gedit 应用程序中。添加插件的具体方法为：选择“编辑”|“首选项”菜单命令，弹出“Gedit 首选项”对话框，单击“插件”选项卡，在“插件”选项卡中罗列了 Gedit 应用程序所支持的插件，如图 12-8 所示。



图 12-8 插件

选中“插入日期/时间”复选框，激活此插件。激活的插件菜单都会作为子菜单从属于“编辑”菜单或“工具”菜单。例如“插入日期/时间”插件已激活，与其相应的菜单“插入日期和时间”显示在“编辑”菜单下。因此选择“编辑”|“插入日期和时间”菜单命令，就可以使用“插入日期和时间”插件了。

12.1.4 高亮显示

Gedit 的一大特点就是高亮显示文本。在编辑文件的时候指定相应的突出显示模式有利于编辑文本，提供编辑效率。

在 Gedit 中，用户可指定文本以何种模式突出显示，突出显示的文件格式包括纯文本、源代码、脚本、标记语言、学科和其他。源代码显示包括我们熟悉的 C++、Java 和 sql 等。标记语言包括 html、xhtml、xml 等，脚本包括 JavaScript、php、sh 等。选择“查看”|“突出显示模式”菜单命令，将显示 Gedit 支持的突出显示模式如图 12-9 所示。



图 12-9 指定突出显示模式



选中“标记语言”子列表下的“HTML”选项，那么文本就会以 HTML 标记语言的语法来突出显示文本。

12.2 办公软件包 OpenOffice.org



办公套件 OpenOffice.org 是 Ubuntu 中默认的办公套件，它是一个功能强大并且开放源代码的办公套件，它包含以下一些办公应用软件。

- **OpenOffice.org Writer:** 文字处理软件，能使用模板、浏览多种格式的文档，可在文档中插入图像。
- **OpenOffice.org Calc:** 电子表格软件，能够兼容 Microsoft Excel 等多种电子表格的格式。
- **OpenOffice.org Impress:** 演示制作软件，包含许多幻灯片特效，能够将演示文稿保存为 Microsoft PowerPoint 等文档格式。
- **OpenOffice.org Draw:** 绘图软件，能够创建、编辑和排列对象，拥有合并多种图片格式到同一张图片的功能。
- **OpenOffice.org Math:** 数学公式软件，能够编辑数学公式，并输出为 pdf 格式的文档。
- **OpenOffice.org Base:** 低端数据库软件，能够导入和导出 dBase 数据库的数据，能够创建报表。用户可以通过它轻松链接到 MySQL 或 Oracle 等外部关系数据库，建立数据库查询、报表、窗体或 MIS 管理系统等。

在日常生活中，我们使用比较多的软件是 Writer、Calc 和 Impress。因此，本节主要讲述这几个常用的软件。

12.2.1 启动 OpenOffice.org 套件

在 GNOME 桌面的顶部面板上，选择“应用程序”|“办公”|“OpenOffice.org”菜单命令启动 OpenOffice.org，如图 12-10 所示。OpenOffice.org 套件启动后的主界面如图 12-11 所示。



图 12-10 选择“OpenOffice.org”菜单



图 12-11 OpenOffice.org 主界面

在 OpenOffice.org 主界面上，可选择所需使用的软件并单击图标启动相应的软件。例如，启动 OpenOffice.org Writer，可单击“文本文档”按钮。

除了通过主界面启动 OpenOffice.org 套件里的软件外，用户还可在图 12-10 的“办公”菜单列表中，选择其中的菜单项，就可启动相应的套件软件。例如，启动 OpenOffice.org Writer，可选择“OpenOffice.org 文字处理”菜单命令。

技术提示

读者在使用 Ubuntu 时，“办公”菜单列表可能并不包括“OpenOffice.org”和“OpenOffice.org 公式”的菜单。读者可参考第 7 章的 7.3.1 小节，在“主菜单”首选项中设置显示 OpenOffice.org 套件组成软件的菜单。简单操作步骤为：在 GNOME 桌面的顶部面板上选择“系统”|“首选项”|“主菜单”菜单命令；然后在“主菜单”对话框中，选择“办公”选项，在罗列出的菜单中，激活“OpenOffice.org”和“OpenOffice.org 公式”；然后关闭“主菜单”对话框。

12.2.2 OpenOffice.org Writer

OpenOffice.org Writer 是 OpenOffice.org 套件的一部分，专门负责文字处理功能，类似于 Microsoft Office 中的 Word。OpenOffice.org Writer 可以读取大多数流行的办公文件格式，可以在保证兼容性的前提下轻松转换其他办公套件的文件。Writer 拥有强大的文字处理能力和文档排版功能，能够让用户自行绘图，提供专业表格的计算功能。接下来讲述 Writer 的基本操作，包括：基本文本处理操作、文字排版操作、插入图片操作、插入表格操作、编辑页脚页眉、页面设置和打印。

1. 基本文本处理操作

选择顶部面板“应用程序”|“办公”|“OpenOffice.org 文字处理”菜单命令，启动 Writer，其主界面如图 12-12 所示。



图 12-12 Writer 主界面

启动 Writer 之后，将鼠标光标移动到 Writer 的文本显示框中，从键盘上输入一首诗“相思”，其步骤为：输入“相思”字符串，然后按下回车键换行，接着按照同样的方法，依次输入“王维”、“红豆生南国，春来发几枝。”和“愿君多采撷，此物最相思。”字符串，如图 12-13 所示。



将文本输入到文件中，现在需要保存文档，以便下次打开文档时，文本还存在。文件保存的方法为：选择“文件”|“保存”菜单命令或单击工具栏上的“保存”按钮，弹出“保存”对话框，如图 12-14 所示。在“保存”对话框的“名称”输入栏上输入文件的名称“test”，然后指定保存文件的文件夹为主目录，单击“保存”按钮，将文件“test.odt”保存为默认的 ODF 文本文档格式，并保存在主目录下。



图 12-13 输入文字



图 12-14 保存新建文件

补充说明

在保存 OpenOffice.org writer 文件时，可在“保存”对话框中设置文件保存的文件格式。单击“保存”对话框中的“文件类型”按钮，就可选择 OpenOffice.org 所支持的文件格式类型。OpenOffice.org writer 支持的文件格式包括微软 Office 的 Word 97/2000/XP 文档格式、OpenOffice.org 1.0 版本的文档格式、文本格式、html 网页格式和 xml 文档格式等。

2. 文字排版操作

OpenOffice.org Writer 提供了强大的文字排版功能，包括字体字号的设置、文体段落的设置和样式的使用等。接下来对上一小节保存的 test.odt 文件中诗词“相思”的内容进行如下排版，具体操作如下所示：

(1) 修改“相思”两字的样式、字体、字号和对齐方式。按下鼠标左键然后拖动鼠标，选中“相思”两个字。单击格式工具栏上的“样式和格式”图标，弹出“样式和格式”对话框如图 12-15 所示，在底部的下拉列表中选择“文本样式”选项，然后双击“标题 1”选项。单击格式工具栏上的字体名称下拉列表，从列表中选择“FreeSerif”字体，单击其后的字号大小下拉列表，从列表中选择“三号”选项，再单击文本的对齐方式“居中”按钮，操作结果如图 12-16 所示。



图 12-15 样式和格式

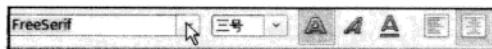


图 12-16 字体、字号和对齐方式的设置

(2) 修改“王维”文本的段落边框和背景。按下鼠标左键然后拖动鼠标，选中“王维”两个字内容。然后选择“格式”|“段落”菜单命令，弹出“段落”对话框，切换到“边框”选项卡，设置“线条排序”中的底部线条，将线条的风格设定为 0.50pt，选择线条的颜色为“蓝绿色”，设置如图 12-17 所示。切换到“背景”选项卡，从背景颜色列表框中选择“浅灰色”选项，可在右边的预览区看到效果，如图 12-18 所示，单击“确定”按钮完成设置。



图 12-17 设置段落的边框

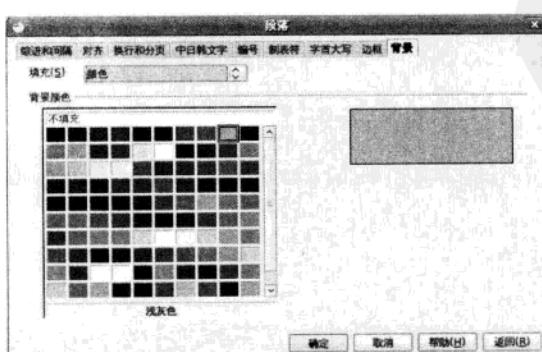


图 12-18 设置段落的背景



(3) 修改“红豆生南国，春来发几枝。”文本的字体颜色，并使其居中对齐。按下鼠标左键然后拖动鼠标，选中“红豆生南国，春来发几枝。”文本内容。然后选择“格式”|“字符”菜单命令，弹出“字符”对话框，切换至“字符效果”选项卡，在字体颜色下拉列表中选中“浅红色”选项，如图 12-19 所示，单击“确定”按钮完成设置。单击格式工具栏上的对齐方式“居中”按钮，使文本内容居中显示。

(4) 修改“愿君多采撷，此物最相思。”诗句的显示格式和上一句的显示格式相同。这里使用一种方便的工具——格式刷。首先按下鼠标左键然后拖动鼠标，选中“红豆生南国，春来发几枝。”诗句，然后单击标准工具栏上的格式刷图标，此时鼠标指针进到工作区里变成油漆桶图标。在工作区中，按下鼠标左键然后拖动鼠标，选中“愿君多采撷，此物最相思。”诗句，在松开鼠标的同时，可发现此诗句的样式和前一句诗句的样式相同。

(5) 完成对“相思”诗词的文字排版，可查看到处理后的结果如图 12-20 所示。

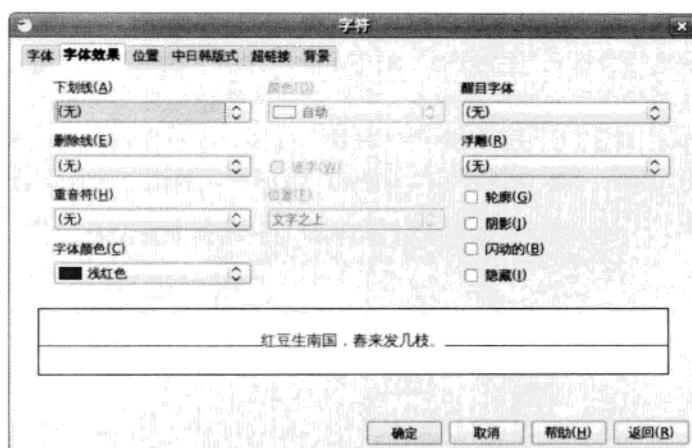


图 12-19 设置字体的颜色

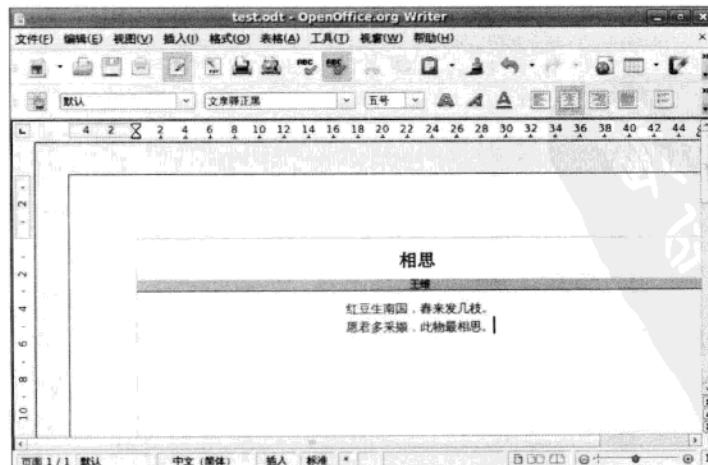


图 12-20 排版后的“相思”诗词

补充说明

在“段落”对话框中，OpenOffice.org Writer除了可设置段落的边框和背景，还提供了设置段落与段落之间的间隔、段落内容的对齐方式、段落内容的缩进方式和行距大小等功能。在“字符”对话框中，OpenOffice.org Writer允许用户设置字体、字号、字体样式、字体效果、文本的位置与旋转角度、超链接和文本的背景。段落的背景可是颜色也可是图形，而字符中的背景只能是颜色。

3. 插入图片操作

在编辑文档的时候，有时为了使文档更加美观，我们会为文档添加必要的图片。而且添加图片更能表达文档的主题或文章的中心。下面我们为“相思”这首诗词添加一幅图片，具体的操作步骤如下所示：

(1) 选择“插入”|“图片”|“来自文件”菜单命令，弹出“插入图片”对话框，在对话框中找到保存在磁盘中的图片“红豆.jpg”，然后单击“打开”按钮，插入图片，图片添加到工作区，如图 12-21 所示。

(2) 单击框架工具栏上的“穿越环绕”按钮，将图片浮在文字的上方。然后设置图片的透明度，在图片工具栏上的“透明”文本框中输入 72%。

(3) 将鼠标移到图片边缘的 8 个小方框上，拖动这些小方框手柄就可以改变图片大小。将鼠标移到已选中的图片上方，指针的图标会变为手状图标，此时按下鼠标左键并拖动鼠标，可在工作区的范围内移动图片。调整合适大小和图片的位置，获得最后的调整效果如图 12-22 所示。

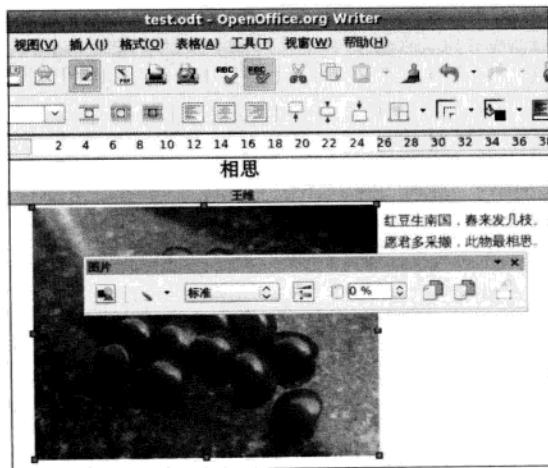


图 12-21 插入图片

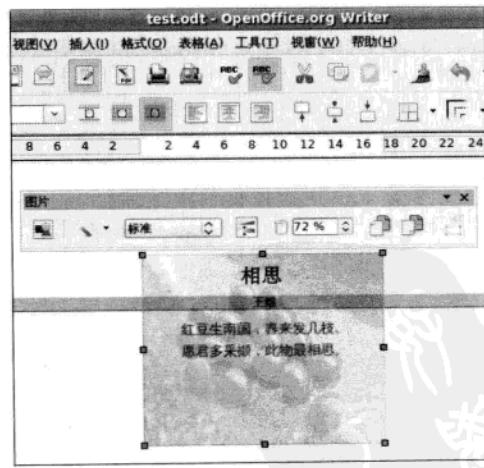


图 12-22 调整图片

双击图片或选择“格式”|“图片”菜单命令，弹出“图片”对话框，如图 12-23 所示。

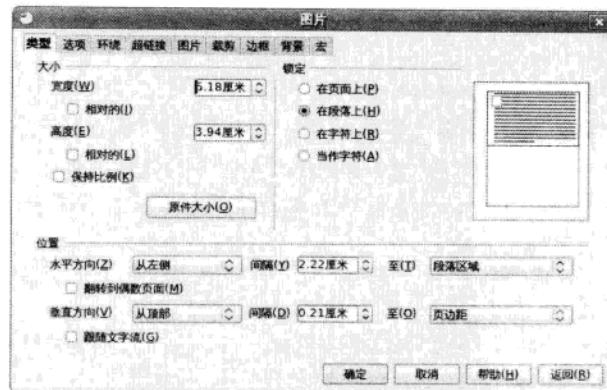


图 12-23 “图片”对话框

在“图片”对话框中，可对图片进行更深入的调整。

补充说明

在调整图片的尺寸时，为了保证图片的长和宽的比例不变，在拖动鼠标的同时按下键盘上的“Shift”键。OpenOffice.org Writer 还提供了丰富的图像处理功能，包括：对图片使用曝光等效果、裁剪图片和编辑图片的轮廓等处理功能。

4. 表格操作

在文档中，使用表格有利于文档内容的组织。在表格的每个单元格中，用户可放置文本和图片，还可以内嵌表格。下面我们使用表格制作一份课表，具体的操作步骤如下所示：

(1) 选择“表格”|“插入”|“表格”菜单命令，弹出“插入表格”对话框，在“名称”输入栏上输入“课表”，其他的设置不变，单击“确定”按钮，向文档工作区插入表格如图 12-24 所示。界面上出现了两行两列的表格，以及表格工具栏。

(2) 合并单元格。按下鼠标左键，拖动鼠标选择第一行的两个单元格，然后单击表格工具栏上的“合并单元格”按钮。合并单元格之后，在第一行单元格中输入“课表”字符串，单击格式工具栏上的“居中对齐”按钮，操作结果如图 12-25 所示。

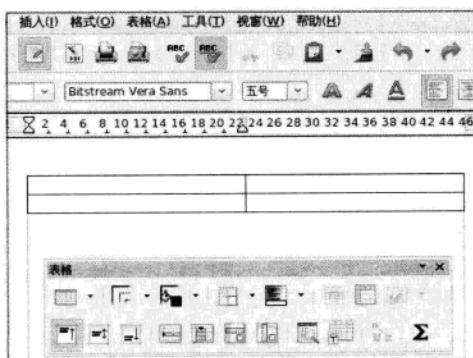


图 12-24 插入表格

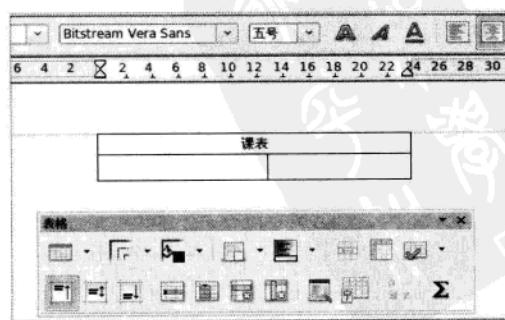


图 12-25 合并单元格

(3) 拆分单元格。拖动鼠标选择表格中的第二行单元格，然后单击表格工具栏上拆分单元格的图标，在弹出的“拆分单元格”对话框中，输入将单元格拆至数目为4，选中方向为垂直方向，单击“确定”按钮，关闭“拆分单元格”对话框，拆分结果如图12-26所示。

(4) 插入单元格。将光标放置在所要添加的行中，然后单击表格工具栏上的插入单元格图标，则在光标所在行的下面添加一行新行。再单击两次，添加两行。然后在单元格中输入相关的课程信息，课表制作的结果如图12-27所示。

在编辑单元格里的内容时，用户不可能一次性编辑成功，其中必然会使用到删除单元格内容、删除单元格、删除表格的操作。

删除单元格内容，可使用“退格”键或“Delete”键。删除单元格，可单击表格工具栏上删除单元格的图标，包括删除行单元格和删除列单元格。若要删除表格，则需要选择“表格”|“删除”|“表格”菜单命令。



图12-26 拆分单元格

时间	一	二	三	四	五	六	日
上午	编程				编程		
下午		编程		编程			
晚上					操作		

图12-27 完成的课表

补充说明

在OpenOffice.org Writer中，用户还可设置表格的边框、表格的背景颜色、边框线条的大小与颜色等，还可使用汇总的功能，以及对表格使用OpenOffice.org提供的表格样式。

5. 编辑页脚页眉

页眉页脚是文档中每个页面里上、下页边距界定的用于存储和显示文本、图形的信息区。现在为上一小节的课表添加页脚页眉。

- 添加页脚

添加文档页脚的具体操作步骤为：选择“插入”|“页脚”|“默认”菜单命令，那么在文档的底部就会出现空白输入框。在输入框中输入“第页”，然后单击格式工具栏上的居中对齐图标使文本居中对齐。将光标放置在“第”和“页”之间，然后选择“插入”|“字段指令”|“页码”菜单命令，插入页码字段，如图12-28所示。

- 添加页眉

添加文档页眉的具体操作步骤为：选择“插入”|“页眉”|“默认”菜单命令，在文档的



顶部就会出现空白输入框。在输入框中输入“2009~2010 学期课表”，然后单击格式工具栏上的居中对齐图标，使文本居中对齐，如图 12-29 所示。

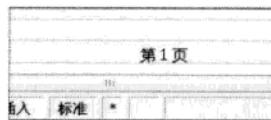


图 12-28 页脚

am Vera Sans	五号	A	A	A
12	14	16	18	20
22	24	26	28	30
32	34			
2009~2010 学期课表				
课表				
时间	一	二	三	四
周	编程			编程
上午				
下午	编程		编程	
晚上				操作

图 12-29 页眉

补充说明

页眉页脚常用于文档的打印修饰，内容可以包括文档当前页页码、文档页数、文档建立日期时间、文档作者、文档标题、文档路径和文件名、公司徽标、边框线、特制图形等文字或图形，这些内容都可在“插入”|“字段指令”子菜单列中找到相应的菜单。

6. 页面设置和打印

文档的打印功能在办公室是最常被使用的功能。在 OpenOffice.org Writer 中，文档可打印输出保存于磁盘的 pdf 文档，还可打印输出到纸张。

若要将文档输出为 pdf 文档，可单击标准工具栏上的“直接输出成 PDF”图标，弹出“导出”对话框，指定 pdf 文档的名称和保存的位置，单击“保存”按钮，就可将文档保存为 pdf 文档格式。

若要将文档打印到纸张，在打印之前需要设定打印的纸张和指定打印机。假设打印机已安装到电脑上，并且打印机的名称为 Samsung-SCX-4200，我们需要做的是在 Writer 应用程序中设置好文档打印的纸张和打印的参数，具体操作步骤如下所示：

(1) 设置打印机。选择“文件”|“打印”菜单命令，弹出“打印”对话框，如图 12-30 所示，在“名称”的下拉列表中选择“Samsung-SCX-4200”选项，然后单击“属性”按钮，弹出“Samsung-SCX-4200 的属性”对话框。在“Samsung-SCX-4200 的属性”对话框中选择“纸张”选项卡，将“纸张格式”设置为“A4-210×297mm”，将“进纸方向”设置为“Portrait (纵向)”，其他参数保持默认值，如图 12-31 所示，单击“确定”按钮完成打印机的操作。

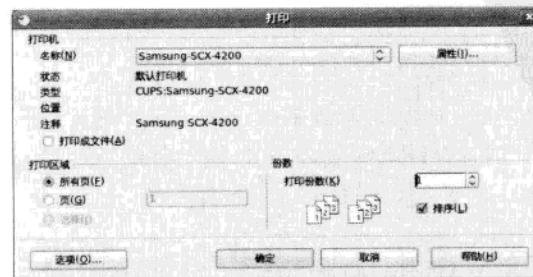


图 12-30 “打印”对话框

(2) 设置打印的参数。在“打印”对话框中选中“打印区域”选项框中的“页”选项，此时若要打印1、3、5页，那么在输入框中输入“1,3,5”字符串。设置“打印份数”的值为2，默认情况下为1。设置完毕之后，单击“确定”按钮，开始进行打印。

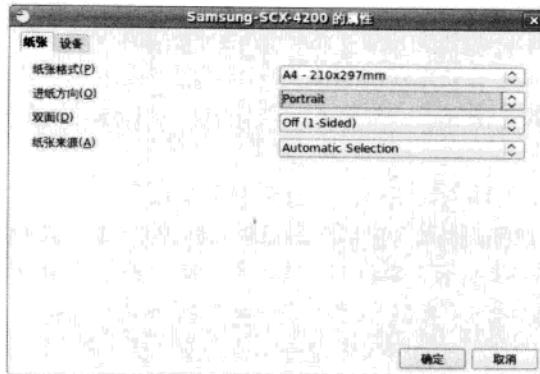


图 12-31 设置打印机

补充说明

只要电脑上安装了打印机并且已设置好打印机，那么用户就可直接使用打印机进行打印文档的操作。一般打印机的进纸方向为Portrait(纵向)。Writer还提供了打印预览的功能，这样用户可在进行打印操作的时候观看文档的打印效果。单击工具栏上的“页面视图”图标，就可查看文档的打印效果。

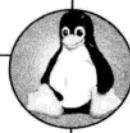
12.2.3 Calc

OpenOffice.org Calc是OpenOffice.org办公套件中专门制作电子表格的应用程序，类似Microsoft Office中的Excel电子表格程序，其方便的数据录入功能以及强大的数据统计功能使其拥有广泛的用户群体。

选择顶部面板“应用程序”|“办公”|“OpenOffice.org 电子表格”菜单命令，启动Calc，如图 12-32 所示。



图 12-32 Calc 主界面



1. 输入数据

要在单元格中输入数据，需要使用光标键指定该单元格，或者单击该单元格，然后在其中输入数据，完成后按回车键。用户还可以使用 Tab 键或“Shift-Tab”组合键移动到右侧或者左侧的单元格。

Calc 中还提供了数据自动补全功能。这一功能大大减少了录入数据的工作量，它会根据当前单元格中的值预测下一个单元格中的值。这一功能不仅适用于数值数据，还适用于一周之内的星期数、一年中的月份等。一般说来，关联到一系列连续整数的任何数据都可以通过自动补全输入。

补全星期一到星期日的输入，具体的操作步骤如下所示：

(1) 在任意一个单元格中输入“星期一”，然后按回车键。此时字符串“星期一”所在的单元格为 B4，如图 12-33 所示。黑框突出 B4 单元格，表示此时 B4 单元格已被选中。

(2) 将鼠标移动到黑框右下角的小黑方块，然后按下鼠标左键，向下拖曳小黑方块，那么 B5~B10 单元格就会分别被自动填入了星期二到星期日，如图 12-34 所示。到星期日数据显示的时候松开鼠标左键，完成操作。

	A	B	C
1			
2			
3			
4		星期一	
5			
6			
7			

图 12-33 输入“星期一”

	A	B	C
1			
2			
3			
4		星期一	
5		星期二	
6		星期三	
7		星期四	
8		星期五	
9		星期六	
10		星期日	
11			

图 12-34 自动补全星期数

这种补全功能除了可向下补全外，还可向上、向左和向右补全。

2. 添加公式

OpenOffice.org Calc 中预定了一些已定义好的函数，在 Calc 的公式中使用函数可提高公式的计算效率。公式或函数在被使用的时候都必须在单元格的开始插入“=”符号，否则会被认为是“静态”数据。

公式一般都是以代数式的形式给出。如“=3*A3+4*(A5+C4/B2)”，其表示将 C4 单元格的值除以 B2 单元格的值，将所得的商加上 A5 单元格的值，再将所得的和乘以 4，所得的值加到 A3 单元格的值与 3 的积。这样可构造比较复杂的公式来完成大量数据的处理工作。

为了简化操作，OpenOffice.org Calc 内置了日期时间、数学、统计、金融、逻辑和其他种类的函数，而且 OpenOffice.org 还设计了一个操作简单的函数向导程序让用户快速生成或编辑公式。使用函数向导对一些数据进行求和处理，其具体操作步骤如下所示：

(1) 在工作区中选择一个空白单元格，然后单击工具栏上的函数向导图标 ，弹出“函数向导”对话框，选择“函数”选项卡，在“分类”下拉列表中选择“数学”选项，在“函

数”选项列表中双击“SUM”选项，如图 12-35 所示。

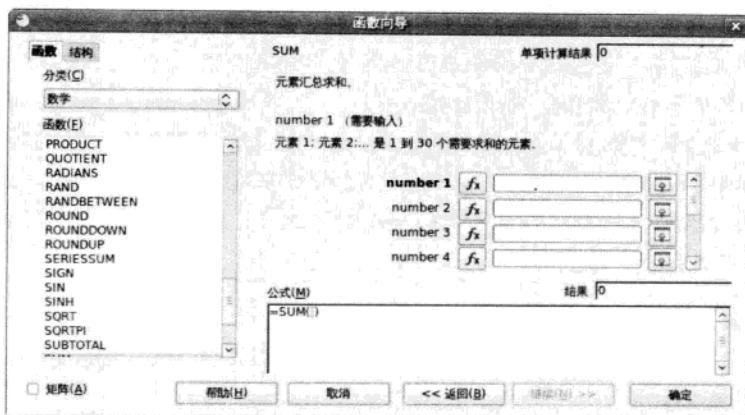


图 12-35 SUM 函数

(2) 单击“number1”输入栏后的缩小 \square 按钮，函数向导对话框缩小，拖动鼠标选择需要处理的数据，单击数据列中的第一个单元格 B4，按住鼠标左键不放，拖动鼠标到数据列的最后一个单元格 B12，松开鼠标，此时“B4:B12”字符串已显示在“函数向导”对话框中，如图 12-36 所示。

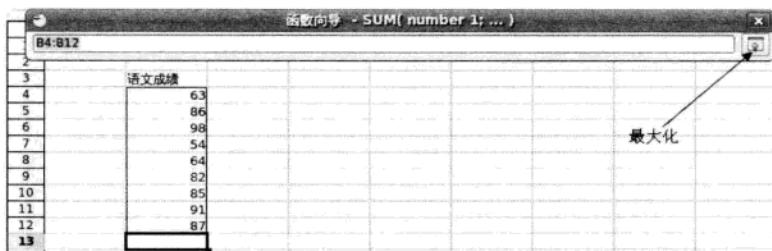


图 12-36 选择需要处理的数据

(3) 数据选择完毕之后，单击“函数向导”对话框中的“最大化”按钮，最大化“函数向导”对话框，此时对话框中所要处理的数据区域已在 number1 的输入栏中。单击对话框中的“确定”按钮，完成对 SUM 函数的调用。

(4) 完成函数调用之后，此时 B13 单元格中的数值为 B4 到 B12 单元格值的总和。选中 B13 可查看到此单元格中的值并不是 710，而是“=SUM(B4:B12)”这样的公式，这可以在工具栏上看到，如图 12-37 所示。

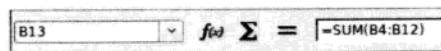


图 12-37 B13 单元格中的内容



补充说明

在 Calc 中，使用 A~Z 的排列数来表示电子表格的列数，如 A.....Z、AA、AB 等，最大是 AMJ 列；而使用 1~9 的排列数来表示电子表格的行数，如 1、2、3 等，最大是 65536 行。在表示每个单元格的时候，字母在前表示列，数字在后表示行，例如，B24 表示第 B 列，第 24 行的单元格。在使用 SUM 函数的时候，我们可以看到数据选区的表示法“B4:B12”。 “B4:B12” 表示数据是从单元格 B4 到单元格 B12 的数据区域。这一数据区域的表示法其实是符号“：“两旁单元格的列与行所形成的一个数据区域。如“C5:D13”区域的数据包括“C5:C13”和“D5:D13”的数据。

3. 插入图表

使用图表可将数据更直观地显示出来，使得用户更容易理解。OpenOffice.org 向用户提供了多种图表的模板。

有数据才能通过数据生成数据图表，因此在插入图表之前一定要准备好数据，也就是需要选择数据区域。假设目前 Calc 中已有一张成绩表，拥有“编号”、“数学成绩”、“语文成绩”和“英语成绩”栏目，为这张成绩表生成一张图表以分析学生的综合成绩以及查看偏科现象。生成图表的具体操作步骤如下：

(1) 选中所要处理的数据区域。在所要处理的数据区域中，在左上角的单元格 A1 上按下鼠标左键，然后拖动到右下角的单元格 D16，所选的数据区域显示为淡黄色，如图 12-38 所示。

A1:D16		f(x)	Σ =	90	
	A	B	C	D	E
1	编号	数学成绩	语文成绩	英语成绩	
2	1	62	63	85	
3	2	76	86	65	
4	3	86	98	90	
5	4	90	54	87	
6	5	65	64	88	
7	6	76	82	89	
8	7	75	85	65	
9	8	80	91	95	
10	9	98	87	61	
11	10	63	64	86	
12	11	77	87	66	
13	12	87	99	91	
14	13	91	55	88	
15	14	66	65	89	
16	15	77	83	90	
17					
18					

图 12-38 选择数据区域

(2) 选中数据区域之后，单击“插入”|“图表”菜单命令，弹出“图表向导”对话框，选择左边“步骤”列表中的“1.图表类型”选项，在右边的“选择一种图表类型”列表中选择“网状图”选项，如图 12-39 所示。

(3) 单击“2.数据区域”步骤，选择“第一列作为标签”选项。单击“4.图表元素”按钮，在“主标题”输入栏中输入“综合成绩分析图表”字符串，如图 12-40 所示。



图 12-39 设置图表类型

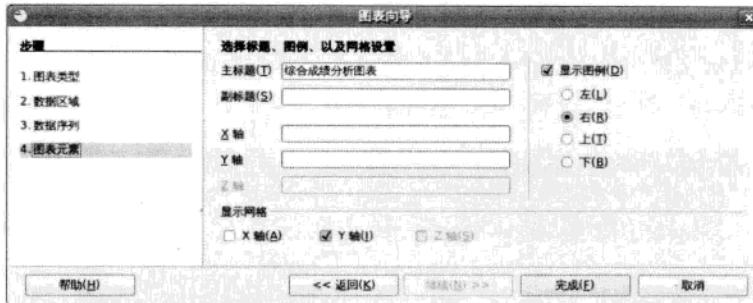


图 12-40 设置图表元素

(4) 单击“完成”按钮，完成图表的生成，如图 12-41 所示。

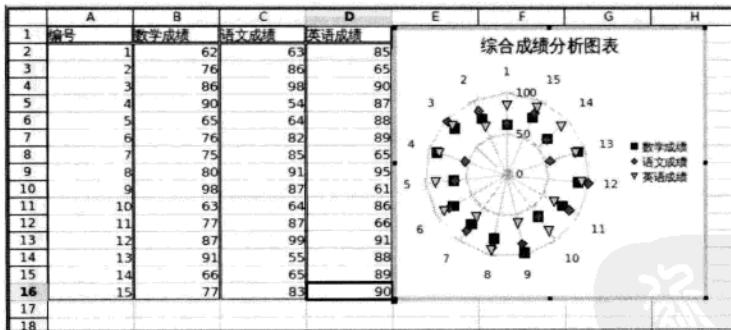
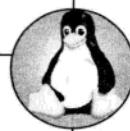


图 12-41 综合成绩分析图表

补充说明

电子表格中的图表是“动态”的，这意味着修改单元格中数据的时候，图表将会进行自动更新。生成图表之后，用户可双击图表中不同的图表元素以修改图表元素的属性，也可在格式工具栏上修改图表类型等。



12.2.4 Impress

OpenOffice.org Impress 是 OpenOffice.org 办公套件中制作幻灯片的应用程序，类似于 Microsoft Office 中的 PowerPoint 应用程序，其操作简单、使用方便，可制作出具有个性化的幻灯片，它是特为制作教学、报告等方面的幻灯片而设计的。Impress 应用程序提供了多种幻灯片模板，还预置了各种各样的动画效果，并拥有多种播放方式。

1. 制作幻灯片

选择顶部面板“应用程序”|“办公”|“OpenOffice.org 演示”菜单命令，启动 OpenOffice.org Impress。如果是首次启动 OpenOffice.org Impress，系统就会出现“演示文稿向导”对话框，这一演示文稿向导可以帮助用户从已有的样式模板集合中创建演示文稿，也可引导用户打开已有的演示文稿。使用演示文稿向导创建演示文稿的具体操作步骤如下所示：

(1) 在“演示文稿向导”对话框中，选择“来自模板”选项，然后在“演示文稿”的选项框中选中“Green”的样式模板，在预览窗口中可查看到效果，如图 12-42 所示。

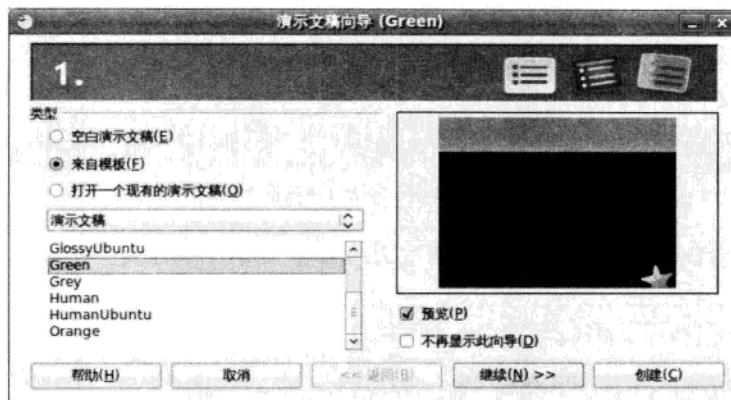


图 12-42 演示文稿向导步骤 1

(2) 单击“继续”按钮，进入向导的下一步，设置幻灯片的背景和输出媒体，还可设置幻灯片切换的快慢，以及切换的方式。如没有特殊要求，使用系统默认设置，单击“创建”按钮，结束向导，进入 Impress 主界面，如图 12-43 所示。

(3) 进入主界面后，在左边的幻灯片面板上显示了唯一的一页幻灯片，这页幻灯片使用了默认的版式。而在主界面右边的任务面板上显示了版式样式列表。通过选择列表中的版式可改变当前幻灯片的版式。在列表中找到名为“标题，剪贴画，文字”的版式，将其应用于第一页幻灯片。

(4) 只有一页的幻灯片不能成为一个完整的演示文稿，这就需要为文稿添加幻灯片。添加幻灯片有多种方法，其中包括：用鼠标右键单击幻灯片面板区域，选择快捷菜单中的“新建幻灯片”|“插入”|“幻灯片”菜单命令。新建幻灯片 2、3，再为它们应用不同的版式，结果如图 12-44 所示。



图 12-43 Impress 主界面



图 12-44 新建幻灯片

在幻灯片的文本框中按照提示输入文字就可编辑幻灯片。

补充说明

在创建演示文稿的时候，母版已设定，但是往往此母版的样式并不一定符合要求，在创建演示文稿之后就需要对母版进行必要的修改。这种修改可通过修改任务面板母版页的样式来完成。母版规定了整个演示文稿的样式，适当的母版可很好地表现主题，吸引观众的注意力。

2. 设置动画

在幻灯片中添加动画可使演示文稿更加生动，使报告或演讲更精彩。在制作幻灯片时可将动画分为幻灯片切换动画和自定义动画。

幻灯片切换动画是 Impress 中预定的针对整张幻灯片而设计的幻灯片切换效果。单击主



界面右边任务面板上的“幻灯片切换”按钮，用户可选择多种动画应用于当前选择的幻灯片，还可设置切换幻灯片的同时播放声音，如图 12-45 所示。单击“应用于选中的幻灯片”列表框中的选项，相应的效果就会作用于当前的幻灯片。



图 12-45 设置幻灯片切换效果

自定义动画可设置幻灯片中每一行字符串的动作。为幻灯片添加自定义动画的具体操作步骤如下所示：

- (1) 在幻灯片中添加图片，输入文字，如图 12-46 所示。
- (2) 使用鼠标拖选“红豆生南国，”字符串，然后单击界面右边的任务面板上的“自定义动画”按钮，在展开的控制面板上单击“添加”按钮，弹出“自定义动画”对话框，选择“入口”选项卡，选择“菱形”选项，如图 12-47 所示，然后单击“确定”按钮，完成自定义动画的添加。

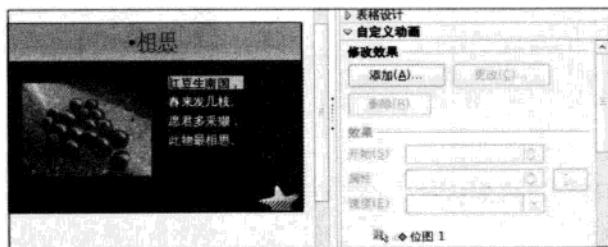


图 12-46 添加图片和输入文字

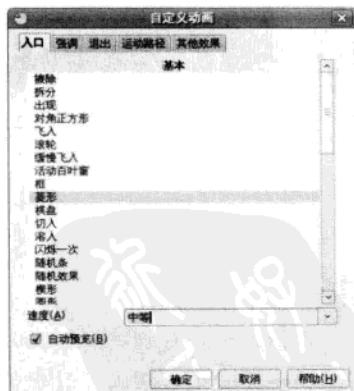


图 12-47 “自定义动画”对话框

- (3) 使用上述的方法，依次为剩余其他的诗句添加自定义动画。添加的自定义动画会显示在任务面板的“自定义动画”列表框中，如图 12-48 所示。

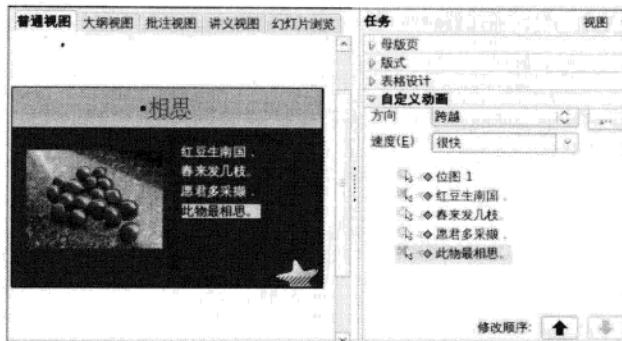


图 12-48 所添加的自定义动画

补充说明

无论是切换幻灯片还是自定义动画，它们都拥有“速度”的属性。在切换幻灯片中速度有3种值：慢、中等和快；而在自定义动画中它还拥有其他2种值：很慢和很快。在设置自定义动画时，用户除了可设置项目进入的动画外，还可设置项目进入动画的行动路径，以及设置项目退出的动画。对已定义的自定义动画，用户可修改动画的执行顺序。

3. 播放幻灯片

幻灯片制作完毕之后，我们可选择“演示文稿”|“幻灯片放映”菜单命令按照默认的幻灯片播放设置，播放幻灯片文稿。单击播放的幻灯片就可激活幻灯片的切换效果或自定义动画，如图 12-49 所示。用鼠标右键单击播放中的幻灯片，弹出快捷菜单，如图 12-50 所示。

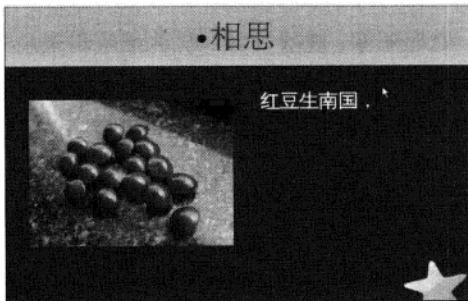


图 12-49 播放幻灯片

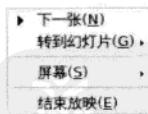


图 12-50 弹出的快捷菜单

演示文稿播放完毕后，幻灯片会自动退出。若用户在播放幻灯片期间需要中断播放，那么按下 Esc 键或选择快捷菜单中“结束放映”菜单命令即可。

除了使用默认设置播放幻灯片，我们还可设置其他方式播放幻灯片。方法为：选择“演示文稿”|“幻灯片放映设置”菜单选项卡，弹出“幻灯片放映”对话框，需要设置显示画笔，那么选择“鼠标显示成笔型”选项，如图 12-51 所示，然后单击“确定”按钮，完成设置。再次播放幻灯片时，鼠标显示为笔型，而且用户可在幻灯片上使用画笔讲解演示文稿。

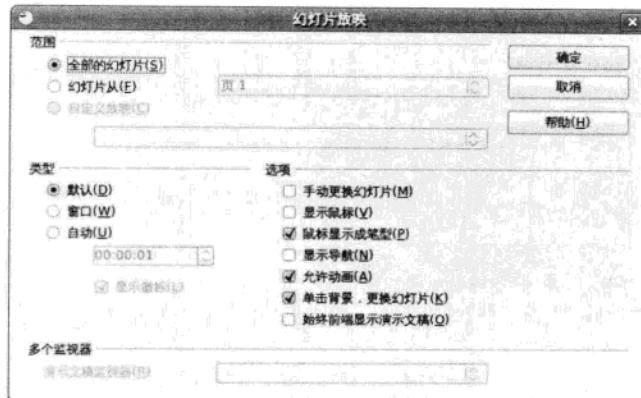


图 12-51 设置幻灯片放映

补充说明

演示文稿制作完毕之后，Impress 提供了导出为其他文件格式的功能。若需要在网络上上传播，用户可将演示文稿导出为 html 格式的文档。若需要使用 Flash 播放，可将演示文稿导出为 swf 的文件格式。但是将演示文稿导出为其他格式，一些数据会被丢失。如导出为 swf 文件格式时，自定义动画无效。

12.3 多媒体软件



Ubuntu 中提供了满足用户观看图片、处理图像、观看视频、欣赏音乐和刻录光盘的要求。使用 F-Spot 观看图片；使用 GIMP 处理图像和设计图形；应用 Totem 电影播放机观看视频；使用 Rhythmbox 播放 mp3 音频欣赏美妙的音乐；使用 GNOME Barker 刻录软件将必要的数据记录到 DVD 中保留下来。

12.3.1 照片管理器 F-Spot

F-Spot 是一套专为 GNOME 桌面环境而设计的个人相片管理软件，它支持所有主要的图像格式，包括 JPEG、PNG、TIFF、DNG 与少数厂商定制的普及细致影像储存格式，如 CR2、PEF、ORF、SRF、CRW、MRW 与 RAF。此外，它还可支持 GIF、SVG 与 PPM 等图片格式。它是源代码开放的软件，拥有简单友好的用户界面，能快速准确地对图像进行颜色调整，包括亮度、对比、色调、明度、温度等，并且支持非破坏性的多版本照片编辑。

F-Spot 可方便地从数码相机中导入数码图像，还可导入保存在磁盘中的图像。F-Spot 提供基本的图片编辑功能，如裁剪、红眼消除、降低饱和度、软焦距和颜色调整等。

选择顶部面板的“应用程序”|“图形”|“F-Spot 照片管理器”菜单命令，启动 F-Spot 应用软件，显示 F-Spot 的主界面，如图 12-52 所示。



图 12-52 F-Spot 的主界面

1. 导入图像

为了方便管理图片, F-Spot 允许用户从数码相机或磁盘等存储介质中导入图像。以导入主文件夹下的 pic 文件夹里的图像为例, 讲述 F-Spot 导入图像的功能, 其具体操作步骤如下所示:

(1) 选择“照片”|“导入”菜单命令, 弹出“导入”对话框, 在“导入源”下拉列表中选择“选择文件夹”选项, 弹出 Import 对话框, 从磁盘中选择保存图片的目录, 然后单击“打开”按钮, 关闭 Import 对话框, 返回到“导入”对话框, 如图 12-53 所示。



图 12-53 已选择图片文件夹下的“导入”对话框

(2) 单击“导入”按钮, 完成图片的导入操作。

2. 编辑图片

数码相机有时并不能拍摄出真实的图像, 如拍摄人物时, 图像中的人物可能出现红眼现象。有时我们拍摄出来的图像光亮度不足, 或是图像没有达到所要的视觉效果。其实这些问题都可使用 F-Spot 来解决。



利用裁剪功能裁剪图像的具体操作步骤如下：

(1) 单击工具栏上的“编辑图像”按钮，选择所要编辑的图像，如图 12-54 所示。



图 12-54 编辑图像的界面

(2) 单击界面左边“编辑”面板上的“裁剪”按钮，然后使用鼠标在图像上拖选裁剪的区域，如图 12-55 所示。

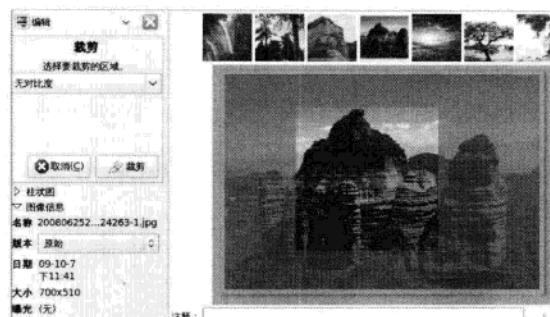


图 12-55 选取裁剪区域

(3) 所选的区域会高亮突出显示，单击右边的“裁剪”按钮，就可将高亮显示区域裁剪为一张新图像，如图 12-56 所示。

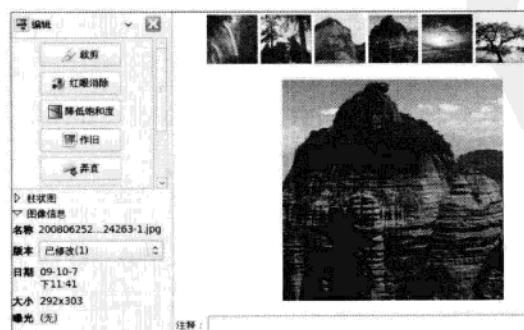


图 12-56 裁剪后的效果

补充说明

F-Spot 会记录图像的编辑版本。比较图 12-55 和图 12-56，可发现“图像信息”面板中的“版本”属性在两图中是不同的。经过裁剪操作后，F-Spot 为所编辑的图像创建一个新的版本，默认命名为“已修改 (1)”，并且以这一版本显示。用户可在“版本”的下拉列表中选择“原始”选项，使图像显示为原始图像。为了方便管理，版本的名称是可修改的，方法为：选择“照片”|“重命名版本”菜单命令，弹出“重命名”对话框，然后输入所要指定的版本名称，再单击“确定”按钮。用户还可删除图像的版本，方法为：选择“照片”|“删除版本”菜单命令，删除图像的版本。F-Spot 使用图像的版本来管理图像，这些版本信息只存在于 F-Spot 软件中。在 F-Spot 中对图像的修改并不会影响到保存在磁盘中的原始图像，也就是说使用其他应用程序打开磁盘中的图像文件时，图像显示的是原始的图像。

12.3.2 图片处理器 GIMP

在 Windows 操作系统中，我们使用得比较多的图形图像处理软件莫过于 Adobe 的 Photoshop 软件了。GIMP 是一款 Ubuntu 附带的处理图形、图像的开源软件，它是 GNU Image Manipulation Program 的缩写。在 GIMP 中创建图像，既可导入图形、图像和 3D 图像，还可以创建一个新的画布进行手动绘制图形。

启动 GIMP 应用程序，可选择顶部面板上的“应用程序”|“图形”|“GIMP 图片编辑器”菜单命令或在终端输入 GIMP&命令，启动后的主界面如图 12-57 所示。

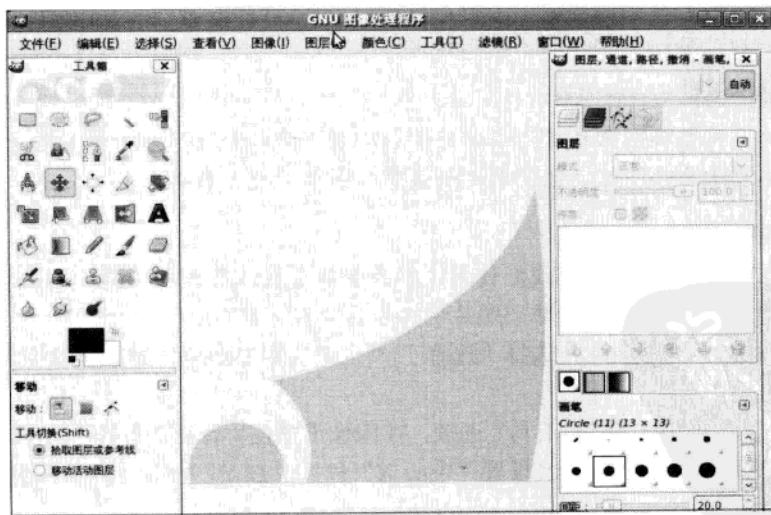


图 12-57 GIMP 主界面

GIMP 界面简洁、实用，其由工具箱、图像窗口和停靠的对话框组成。可停靠的对话框可从“窗口”|“可停靠对话框”菜单中显示和隐藏。

目前，数码相机的使用越来越普及，那么处理数码照片也就成为必然。处理图形、图像



功能强大的 GIMP 也就有了发挥的空间。使用 GIMP 可使数码图像更加美观，可为数码图片添加特效，改变数码图片在颜色上的不足。下面就介绍如何使用 GIMP 处理数码照片。

1. 裁剪数码照片

为了突出照片的主题或是去除不必要的部分，通常要对照片进行裁剪。使用 GIMP 对数码照片进行裁剪操作十分方便，具体的操作步骤如下所示：

(1) 启动 GIMP 后，打开欲裁剪的数码照片，单击 GIMP 工具箱中的“裁剪工具”图标，这时会出现一个可以随意调整大小的矩形方框，按住鼠标左键不放并拖动它，使用此方框在数码照片上准确地选中裁剪区域边缘，如图 12-58 所示。

(2) 通过鼠标移动矩形边缘的小矩形，可调整矩形的大小；单击矩形的其他区域，可调整矩形的位置。调整完毕之后，双击矩形区域，完成图像的裁剪，如图 12-59 所示。

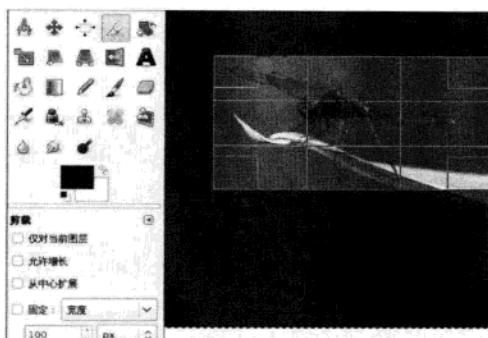


图 12-58 选中裁剪区域

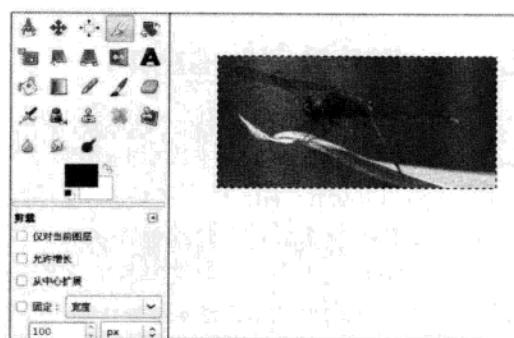


图 12-59 裁剪后的效果

补充说明

裁剪指的是将整幅图像裁剪掉不要的部分，留下的部分就作为新的图像。裁剪后的图像大小总是不大于原始图像，也就是说经裁剪后图像的尺寸会变小。

2. 旋转数码照片

有时拍摄的图像会歪斜，或我们使用歪斜的图像来表达图像的动感，那么 GIMP 可以轻而易举地为我们完成任务。旋转数码照片的操作步骤如下所示：

(1) 使用 GIMP 打开照片，选择“选择”|“全部”菜单命令，选择整幅图像，将图像作为选区。

(2) 单击工具箱中的“旋转工具”选项，在图像上单击，弹出“旋转”对话框，移动“角度”滑动块调整选区的旋转角度，设置“中心 X”和“中心 Y”的值可调节图像的旋转中心，如图 12-60 所示。

(3) 旋转的角度也可通过在图像上拖动鼠标来获得。完成旋转参数的调整之后，单击“旋转”按钮，完成图像的旋转，如图 12-61 所示。



图 12-60 设置旋转参数

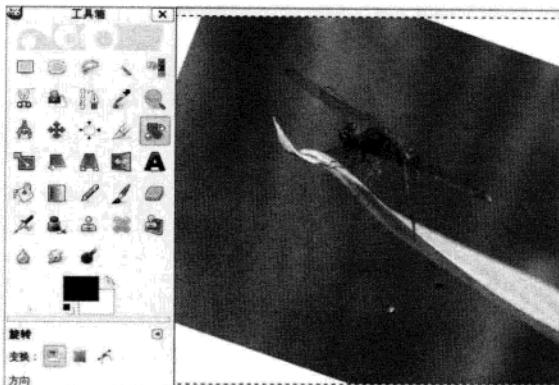


图 12-61 旋转图像后的效果

3. 转换数码照片的格式

使用数码相机 (DC) 拍摄或是用扫描仪扫描的照片通常保存为 JPEG 格式，但由于大家需求不尽相同，因此就产生了图片格式转换的问题，例如要将图片用于出版和印刷时 BMP 或是 TIFF 格式就要比 JPEG 方便得多。GIMP 数码照片支持的图片格式众多，有二十七八种。它不但支持常见的 BMP、JPEG、GIF、PNG，还支持一些比较专业的图像格式，如此它就能很轻易地解决图像格式之间的转换问题。

只需使用 GIMP 打开要转换的图片，在窗口菜单中选择“文件”|“另存为”菜单命令，弹出“保存图像”对话框，简单地修改一下图片文件的扩展名，将 jpg 改为 bmp、tiff 或其他的扩展名，重新保存，即可完成图片格式的转换。单击“保存图像”对话框中的文件类型，会弹出 GIMP 所支持的文件格式，如图 12-62 所示。

4. 调整数码照片的明亮度

数码照片光线不好或过暗、光圈调整欠佳等，都会造成数码照片效果不好。这些数码照片都可以使用 GIMP 进行处理，并且十分方便。例如，有一张海边的照片，由于当时为薄阴天气缘故，照片灰暗缺乏亮度，调整其明亮度的操作步骤如下所示：

(1) 使用 GIMP 打开该海边图片，选择“颜色”|“色阶”菜单命令，弹出“色阶”对话框，如图 12-63 所示。

(2) 在通道下拉列表中，选择“明度”选项。移动“输入色阶”中的三个黑、灰和白的三角形滑块，即可调节图像的明亮度。将白色三角形滑块向左边移动，使更多的像素变得更明亮；而将黑色三角形滑块向右边移动，就可使更多的像素变得更沉暗；而移动灰色三角形滑块则可调节灰色像素的多寡。



图 12-62 GIMP 所支持的文件格式



5. 调整数码照片的色彩

在实际拍摄中，由于环境、时间的不同，数码照片常常出现偏色，例如在室内拍摄的照片中最容易出现偏红色或偏黄色。使用 GIMP 的色相/饱和度来调整照片的偏色，可以使色彩趋于正常、和谐。例如，照片存在深红色偏色，可以打开该图片，从图像窗口的菜单栏中选择“颜色”|“色调-饱和度”菜单命令，弹出“色调-饱和度”对话框，选择“主值”或其他色系，调整“色调-饱和度”对话框上的色调、亮度和饱和度滑块，直到满意为止，如图 12-64 所示。

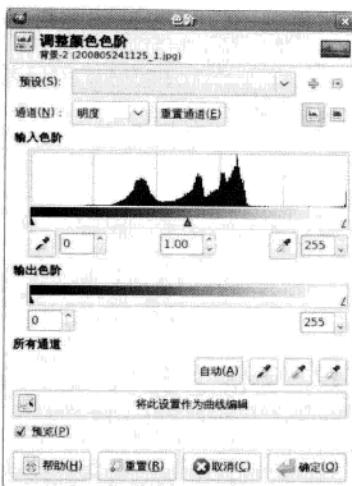


图 12-63 “色阶”对话框

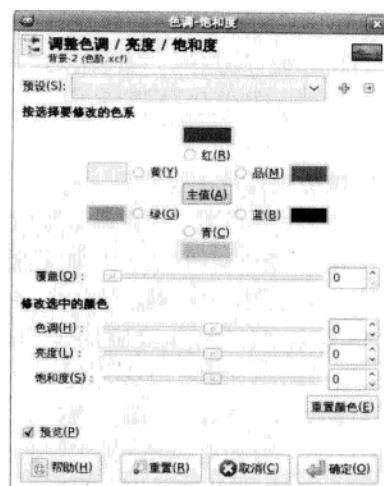


图 12-64 “色调-饱和度”对话框

6. 处理图像模糊的数码照片

拍摄照片时，由于焦距设定不当或拍照时手抖动，而造成照片模糊。此时，可以使用 GIMP 的锐化滤镜来弥补。基本操作方法为：打开由于焦距不准而造成图像模糊的照片，从图像窗口的菜单栏中选择“滤镜”|“增强”|“锐化”命令，弹出“锐化”对话框，在弹出的滤镜参数中设定锐化值，如图 12-65 所示。

移动锐度滑块时，可在预览窗口查看到应用滤镜后的处理效果。



图 12-65 “锐化”对话框

补充说明

GIMP 提供了多种滤镜。滤镜是通过不同的方式改变像素数据，以达到对图像进行抽象、艺术化的特殊处理效果的。而且 GIMP 是开放源码的应用程序，这使得很多软件设计师用 GIMP 在基础上设计更多实用的滤镜。GIMP 中所有滤镜的菜单都包括在图像窗口的菜单栏“滤镜”菜单下。

12.3.3 电影播放机 Totem

Linux 默认安装了 Totem 电影播放机，这一电影播放机是基于 Gstreamer 插件的，它可播放多种媒体格式文件，而在 Ubuntu 9.04 中，Totem 可为视频添加字幕。

1. 播放 rmvb 格式的视频

目前，选择网络上传播的视频多为 rmvb 格式的，视频文件会以 rmvb 或 rm 为后缀。那在 Ubuntu 中怎么播放这种格式的视频文件呢？使用 Totem 可满足用户的要求，而且使用简单，具体的操作步骤如下所示：

- (1) 将视频文件下载到本地磁盘后，用鼠标右键单击视频文件，选择快捷菜单上的“用‘电影播放机’打开”菜单，此时视频文件被打开，如图 12-66 所示。
- (2) 在左下角的控制面板上，移动时间滑块可指定视频播放的时间起点；单击“播放”、“下一章/电影”和“上一章/电影”等按钮可改变视频播放的状态。
- (3) 单击“声音控制”按钮，可调节视频的声音。



图 12-66 播放 rmvb 格式的视频

补充说明

也许第一次使用 Totem 播放 avi、rmvb 等视频的时候，Totem 并不能如愿打开视频文件，这是由于 Totem 基于 Gstreamer 插件，而 Gstreamer 插件并没安装到系统中。此时，我们要安装 Gstreamer 系列的插件。打开新立得软件包管理应用程序，从中搜索含 gstreamer 字符串的软件包。在搜索结果中选择：gstreamer0.10-pitfdll、gstreamer0.10-ffmpeg、gstreamer0.10-plugins-bad、gstreamer0.10-plugins-bad-multiverse、gstreamer0.10-plugins-ugly 和 gstreamer0.10-plugins-ugly-multiverse 软件包。

2. 添加字幕

外国的电影影响十分广泛。在观看外国电影时添加中文字幕也就成为必要的操作了。在 Ubuntu 9.04 中，电影播放机可轻易地为外国视频添加中文字幕，其操作如下所示：



(1) 在射手网 (<http://shooter.cn/>) 上搜索视频的中文字幕，然后把字幕文件保存在磁盘中，以备使用。

(2) 在已打开的视频文件的菜单栏中，选择“查看”|“选择文本字幕”菜单命令，在弹出的“选择文本字幕”对话框中，从磁盘中选择已下载的字幕文件，然后单击“打开”按钮，中文字幕就添加到了视频中。

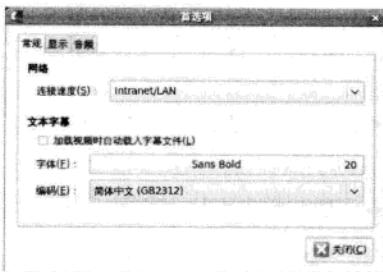


图 12-67 Totem 首选项

(3) Totem 电影播放机默认文本字幕的编码是 unicode 编码格式，而中文字幕文件的编码格式普遍都是简体中文的编码格式，包括 GB2312、HK 和 GBK 等。Totem 设置的文本字幕编码不符合字幕文件的时候，显示在屏幕上的文本就会出现乱码现象。解决字幕乱码的方法为：选择“编辑”|“首选项”菜单命令，弹出“首选项”对话框，选择“常规”选项卡，从编码的下拉列表中选择“简体中文 (GB2312)”选项，如图 12-67 所示。

12.3.4 音乐播放器 Rhythmbox

音乐播放器 Rhythmbox 是 Ubuntu 中默认的音频播放器，它与 Totem 一样也是基于 Gstreamer 插件的。通过 Gstreamer，它可全面支持音频格式，包括 WMA、mp3、AU 和 WAV 等。而且它还支持 CD 音频播放，以及支持搜索和排序库中的音乐。

如同 Totem 一样，在使用它来播放音频之前需要添加 Gstreamer 系列的插件，按照安装 Totem 中添加 Gstreamer 的方法，此次安装所需的软件包，包括 gstreamer0.10-ffmpeg、gstreamer0.10-mpegdemux、gstreamer0.10-gl 和 gstreamer0.10-g nonlin。添加之后就可以顺利地播放音频文件了。

用鼠标右键单击已下载的 mp3 文件，在快捷菜单中选择“用‘Rhythmbox 音乐播放机’打开”菜单命令，打开音频文件，如图 12-68 所示。



图 12-68 Rhythmbox 音乐播放机

Rhythmbox 在 Linux 中使用广泛，它美中不足的是播放 mp3 文件时不能很好地显示中文文件名，出现乱码现象。解决这一乱码问题的方法如下所示：

- (1) 使用新立得软件包管理应用程序安装 python-mutagen 软件包。
- (2) 打开模拟终端，并进入保存有 mp3 文件的文件夹。
- (3) 在终端输入命令：

```
mid3iconv -e GBK */*.mp3
```

这样就可将 mp3 格式中文件的文字编码修改为 GBK 编码格式。重新启动 Rhythmbox 音乐播放机，可观察到原来乱码的文件名等信息显示正常。

12.3.5 刻录软件 GNOMEBaker

之前我们介绍了 Ubuntu 中的默认的刻录软件 Brasero 2.26.1 的使用，接下来将重点介绍 GNOMEBaker 的使用方法。这个软件包是需要自行安装的。

在 Ubuntu 9.04 中，GNOMEBaker 同时支持资料 CD 的刻录和音乐 CD 的刻录。它支持的音乐格式包括 WAV、mp3 以及 OGG3 种。它还可烧录影片 DVD。其主要功能如下：烧录资料光碟、清空可重写光碟、烧录 DVD 光碟、复制资料光碟、复制音乐光碟、支持 multisession 烧录、支持光碟映像档烧录。

目前，网上的精美图像和美妙的音乐不断地冲击着用户的眼睛和耳朵。在欣赏的过程中，可以将它们保存下来，或是将这些图像和音乐与朋友一起共享。使用 GNOMEBaker 可轻松地刻录音频 CD 或 DVD。

在新立得软件包管理器中，找到 GNOMEBaker 软件包，然后执行并安装 GNOMEBaker 软件包。GNOMEBaker 的启动菜单就会添加在主菜单目录下，选择主菜单“应用程序”|“影音”|“GNOMEBaker CD/DVD Writer”命令，启动 GNOMEBaker 应用程序，其主界面如图 12-69 所示。

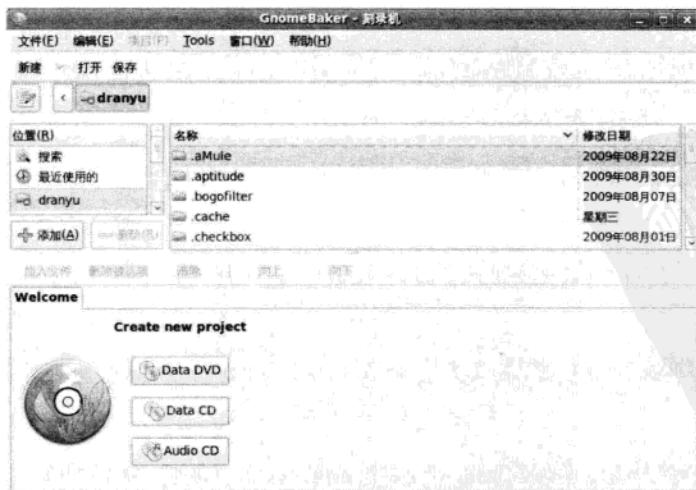


图 12-69 GNOMEBaker 主界面



若要制作一张音频 CD 送给亲朋好友，使用 GNOMEBaker 是个不错的选择。音频 CD 的刻录操作简单，步骤如下所示：

(1) 插入一张空白的光盘到刻录机里，然后单击 GNOMEBaker 主界面上的刻录音频 CD 的按钮 ，从而创建一个新的刻录文件，如图 12-70 所示。

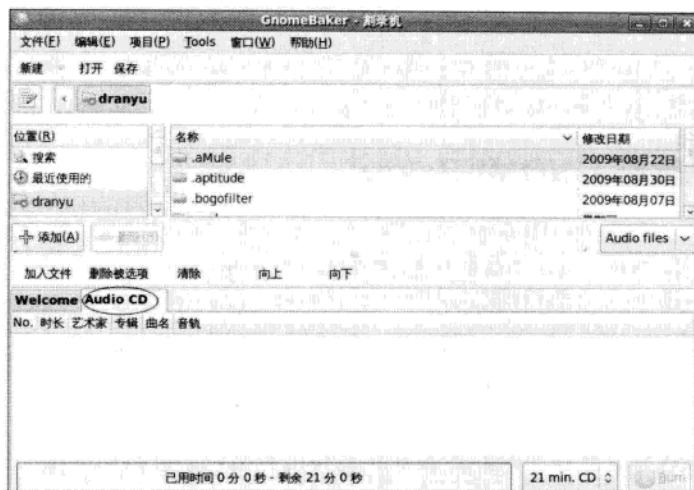


图 12-70 创建音频 CD

(2) 使用鼠标将主界面上文件浏览面板中的音频文件拖到音轨中，如图 12-71 所示。

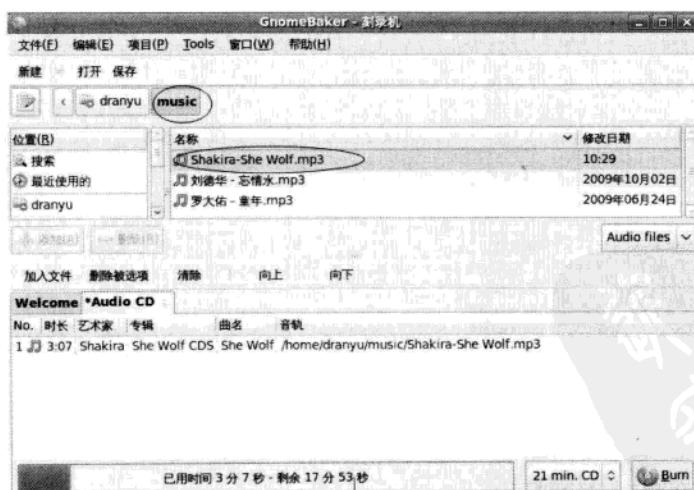


图 12-71 添加音频文件到音轨中

(3) 文件添加完毕后，单击“Burn”按钮 ，弹出“创建音频 CD”对话框，设置速度为 16，选择“安全刻录”选项，如图 12-72 所示。



图 12-72 设置刻录参数

(4) 单击“开始”按钮,GNOME Baker 开始刻录音频 CD。当 GNOME Baker 刻录完毕之后会弹出光盘,这表示我们完成了音频 CD 的刻录。

12.4 即时通信



在网络高速发展的今天,人与人之间的距离越来越近了,即时通信软件的使用将世界各地的人连在一起。目前,比较流行的即时通信软件包括 QQ、MSN、Yahoo Messenger、Skype、AIM 和古老的 ICQ 等,使用这些软件就如同人与人在直接交谈。QQ 在我们国家最为流行,其拥有 8 亿多的网络用户。Linux 下的 Pidgin 通信应用程序集成了多种即时通信软件,可满足用户的多种要求。LibFetion 是 Linux 下的飞信,使得移动用户可与电脑直接通信。

12.4.1 腾讯 QQ

为迎合 Linux 的发展,腾讯公司为 Linux 操作系统开发了 Linux 版本的 QQ 应用程序,用户可从 <http://im.qq.com/qq/linux/> 的官方网上下载。由于我们使用的是 Ubuntu 系统,因而在下载 LinuxQQ 时,需要下载它的 deb 软件包--LinuxQQ_v1.0.2-beta1_i386.deb。

下载 LinuxQQ 的 deb 软件包后,就可以开始安装 LinuxQQ 软件包。在 Ubuntu 中,安装 LinuxQQ 软件包很简单,具体操作步骤如下所示:

(1) 双击保存在磁盘中的 LinuxQQ 软件包,弹出“软件包安装-LinuxQQ”对话框,如图 12-73 所示。

(2) 单击“安装软件包”按钮,开始安装软件包。在询问获取认证身份时,输入 Linux 系统的用户密码。新立得软件包管理器会自动安装 LinuxQQ。

经过上述两个步骤就可完成 LinuxQQ 的安装。安装完毕后,LinuxQQ 的启动项会作为菜单放置在 Ubuntu 的主菜单“应用程序”下的“Internet”下,选择“应用程序”|“Internet”|“腾讯 QQ”菜单命令,启动 QQ 程序,第一次启动时在“账号”栏中输入 QQ 账号,在“密码”栏中输入 QQ 的密码,其他参数保持不变,如图 12-74 所示。然后单击“登录”按钮,登录 QQ 账号。



图 12-73 LinuxQQ 软件包的包信息

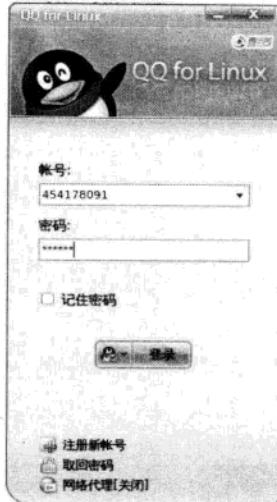


图 12-74 QQ 的登录界面

接着用户就可以和自己的朋友在 Linux 平台上畅谈了。

12.4.2 互联网通信 Pidgin

Pidgin 互联网通信程序是 Ubuntu 中默认安装的即时通信应用程序，是一个跨平台的即时通信客户端，使用 GNU 通用公共许可证发布。它支持多个当前常用的即时通信协议，让用户可以用同一个软件登录不同的即时通信服务，如 QQ、MSN、ICQ 和 AIM 等。

1. 启动 Pidgin

选择主菜单“应用程序”|“Internet”|“Pidgin 互联网通信程序”菜单命令，就可启动 Pidgin。Pidgin 第一次启动会弹出“好友列表”对话框和“账户”对话框。若用户为 Pidgin 添加了账户并设置自动登录，那么启动 Pidgin 后就只显示“好友列表”对话框。

2. 添加 MSN 账户

在没有出现“账户”对话框的情况下，在“好友列表”对话框中选择“账户”|“管理账户”菜单命令，弹出“账户”对话框。以添加 MSN 账户说明 Pidgin 的使用，具体操作如下所示：

(1) 在“账户”对话框中，单击“Add”按钮，弹出“添加账户”对话框，在“协议”下拉列表中选择“MSN”选项，然后输入 MSN 的账户和密码。其他参数可选填，如图 12-75 所示。

(2) 单击“添加”按钮，将此 MSN 添加到了“账户”对话框，如图 12-76 所示。此时 MSN 会登录到网络中，那么“好友列表”对话框中将显示当前登录账户的好友列表信息，如图 12-77 所示。

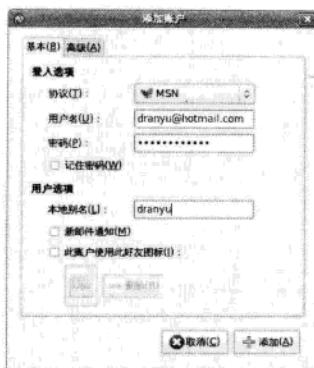


图 12-75 添加账户

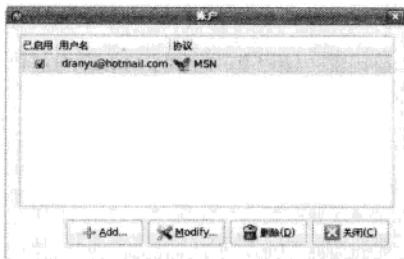


图 12-76 “账户”管理

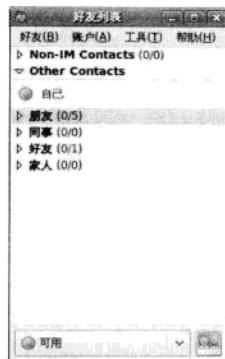


图 12-77 “好友列表”对话框

Pidgin 允许用户添加多个账户，并实现多个账户同时在线。若要启动已添加到“账户”管理对话框中的账户，只要勾选账户前的复选框即可。若要退出账户，同样是勾选账户前的复选框。

在 Pidgin 的“好友列表”对话框中的“工具”菜单中有很多实用的功能，还可为 Pidgin 设置多种插件。

12.4.3 飞信 LibFetion

飞信是中国移动推出的“综合通信服务”即时聊天工具，可以免费从 PC 上发手机短信，而且不受任何限制，实现互联网和移动网间的无缝通信服务。LibFetion 是第三方飞信应用程序，它具有较强的跨平台性，在主流操作系统和手持设备上都有相应的版本。它实现 PC 与手持设备、PC 与 PC 的即时通信。

首先从 http://www.libfetion.cn/Linux_demoapp_download.html 官方网上下载 LibFetion 的 deb 软件包最新版本，保存到磁盘中；然后双击 LibFetion 软件包，进行 LibFetion 的安装。选择“应用程序”|“Internet”|“Linux Fetion”菜单命令，启动 Linux Fetion 应用程序，登录界面如图 12-78 所示。

在“请输入手机号或 Fetion 号”输入栏中输入账户信息，然后再输入账户的密码，单击“登录”按钮即可。Linux Fetion 操作简单，而且界面简洁。

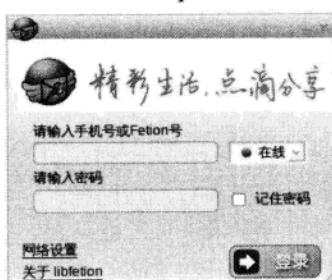
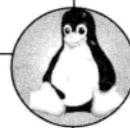


图 12-78 Linux Fetion 的登录界面

12.5 下载工具

目前，网络如此流行，下载工具就成为必不可少的网络工具。Ubuntu 中并没默认安装这一类软件，可它在软件源中已包括了多种下载工具，如 ftp 的下载软件 gFTP、电驴资源下载软件 aMule 等。





12.5.1 gFTP

gFTP 是 XWindow 下的一个用 Gtk 开发的多线程 FTP 客户端工具。它与 Microsoft Windows 下运行的 CuteFTP 等 FTP 工具极为类似，主要有以下一些特性：并行下载；断点续传；传输任务队列；全目录下载；ftp/http 代理传输支持；远程目录缓存；被动/非被动文件传输；文件拖放（Drag and drop）；书签；传输中断等。

在 Ubuntu 下使用 gFTP 之前，需要先安装这一软件。安装方法为：打开新立得软件包管理器，搜索“gftp”，然后选择 gFTP 软件包并执行安装软件包，或在终端输入如下命令：

```
sudo apt-get install gftp
```

这样就可完成 gFTP 的安装。成功安装后选择 Ubuntu 主菜单“应用程序”|“Internet”|“gFTP”菜单命令，启动 gFTP 应用程序，其主界面如图 12-79 所示。



图 12-79 gFTP 的主界面

1. 中文显示远程文件

安装 gFTP 后，程序并不能很好地显示远程文件的中文字符。若要使得 gFTP 能完美显示中文，我们就需要对 gFTP 的配置文件进行修改，具体的修改步骤如下所示：

- (1) 选择菜单栏上的“FTP”|“选项”菜单命令，弹出“选项”对话框。
- (2) 在对话框中的远程字符集输入框中输入如下字符串 cp936、gb18030、gbk、gb2312、utf-8、euc-tw，如图 12-80 所示。
- (3) 设置完毕后，单击“确定”按钮。

2. 连接服务器

在主界面中顶部的连接信息栏上，输入主机名，选择端口，输入用户名和密码，然后按回车键即可。连接成功后，gFTP 会在右边的文件浏览器中显示远程的文件夹和文件，如图 12-81 所示。



图 12-80 “选项”对话框

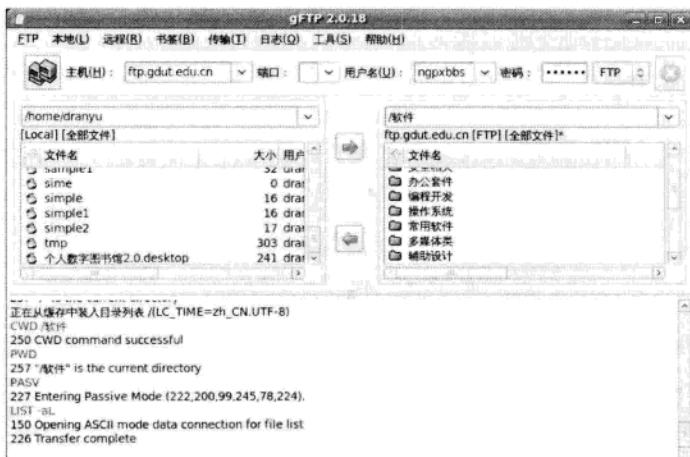


图 12-81 连接服务器后显示的远程文件夹和文件

补充说明

gFTP 下载文件的操作十分简单，只要使用鼠标将右边所需下载的远程文件拖动到左边的本地磁盘上即可。文件拖动后，gFTP 会将指定文件下载到指定的地点。远程文件连接后，用户还可选择“书签”|“添加书签”菜单命令，将目前的 FTP 账号保存下来，方便下次连接的时候使用。

12.5.2 Linux 下的电驴 aMule

在资源共享的网络世界里，电驴是一个完全免费且开放源代码的 P2P 资源下载和分享软件，利用电驴可以将全世界所有的计算机和服务器整合成一个巨大的资源分享网络。用户既



可以在这个电驴网络中搜索到海量的优秀资源，又可以从网络中的多点同时下载需要的文件，以达到最佳的下载速度，用户也可使用电驴快速上传分享文件，达到最优的上传速度和资源发布效率。

电驴 eMule 只有 Windows 的版本，不适合 Linux 系统，而 aMule 类似于 eMule，它是一个开源免费的 P2P 文件共享软件，遵循 GNU 通用公共许可证协议发布，基于 xMule 和 lMule。它经过恰当的设置后可应用 eDonkey 网络协议和支持 KAD 网络。它还可在包括各种类 UNIX 系统、Windows 在内的多种操作系统下运行。

Ubuntu 没有默认安装 aMule 应用程序，安装这一软件的方法为：打开新立得软件包管理器，搜索 aMule 字符串，然后选择 aMule 软件包，并执行安装此软件包即可。安装完毕后，选择主菜单中“应用程序”|“Internet”|“aMule”菜单命令，启动 aMule 应用程序，其主界面如图 12-82 所示。

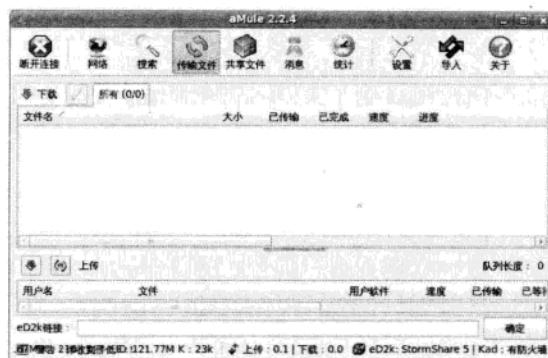


图 12-82 aMule 的主界面

1. 设置文件保存的目录



图 12-83 设置保存目录

为了方便浏览已下载的文件，用户可设置将下载的文件保存到何处，设置的具体方法如下所示：

- (1) 单击“设置”按钮，弹出“设置”对话框。
- (2) 单击左边菜单栏上的“目录”选项，切换到设置目录的界面。
- (3) 单击设置目录界面上“传入”属性后的“浏览”按钮，指定所下载文件的默认保存目录，如图 12-83 所示。

- (4) 单击“确定”按钮，完成设置。

2. 搜索资源

在 aMule 中，aMule 默认设置了许多资源共享的服务器，aMule 可供下载的资源十分庞大。因此，学会使用搜索功能是很重要的。使用 aMule 搜索文件的具体操作如下所示：

- (1) 单击工具栏上的“搜索”按钮，切换到搜索的操作界面。
- (2) 在操作界面的“名字”输入栏，输入“Ubuntu”字符串，选择“开始”菜单命令，aMule 会在资源库中搜索相关的资料。aMule 会将搜索结果显示在界面上，如图 12-84 所示。



图 12-84 搜索文件

补充说明

双击文件记录即可下载指定文件，将文件的下载记录添加到传输文件界面中。或者用鼠标右键单击搜索结果中的文件记录，选择快捷菜单中的“下载”菜单，下载文件。

3. 设置下载 VeryCD 的资源

VeryCD 上的资源多，而资源下载的速度很快，这些优点使很多用户选择了使用它来下载视频、教程和书籍等。在浏览和下载 VeryCD 的资源时，aMule 不会自动添加 VeryCD 上的服务器。因此，也就无法从 VeryCD 上下载所需的资源。设置的具体步骤如下所示：

- (1) 单击工具栏上的“网络”按钮，切换到网络设置界面。
- (2) 添加服务器。在界面上，输入服务器的名称“DonkeyServer No1”，输入服务器的 IP 地址和端口号“62.241.53.2:4242”，然后单击“添加”按钮，添加新的服务器到 aMule 的服务器列表中，如图 12-85 所示。



图 12-85 添加新服务器



(3) 重复第二步，继续添加其他两个服务器，包括：华语 P2P 源动力的 61.152.93.254:4661；Razorback 2 的 195.245.244.243:4661。

(4) 设置 Firefox 网页浏览器，使得 Firefox 可接受 ed2k 协议的文件，并指定使用 aMule 应用程序打开。在 Firefox 的地址栏中输入“about:config”，然后按下“Enter”键，进入 Firefox 的设置界面。在弹出的快捷菜单中选择“新建”|“布尔”菜单命令，创建一个布尔项 network.protocol-handler.external.ed2k，值为 true；然后在快捷菜单中选择“新建”|“字符串”菜单命令，创建一个“字符串”项 network.protocol-handler.app.ed2k，值为 /usr/bin/amule。

(5) 关闭 aMule，重新启动 aMule。关闭 Firefox，重启后，打开 <http://www.verycd.com> 网页即可使用 aMule 下载 VeryCD 上的资源。

12.6 浏览器 Firefox



Mozilla Firefox（以下简称“Firefox”，正式缩写为“Fx”，非正式缩写为“FF”），中文俗称为“火狐”，是由 Mozilla 基金会与开源团体共同开发的网页浏览器。Firefox 是从 Mozilla Application Suite 派生出来的网页浏览器，其拥有标签页浏览、拼字检查、即时书签、下载管理员及自定义搜索引擎等功能。此外，还可以通过由第三方开发者贡献的附加组件来加强各种功能，高扩展性是 Firefox 的特色之一。Ubuntu 默认安装了这一功能强大的网页浏览器。

1. 启动 Firefox

单击 Ubuntu 顶部面板上的 firefox 图标，或者选择顶部面板上的“应用程序”|“Internet”|“Firefox Web Browser”菜单命令，启动 Firefox 应用程序。Firefox 第一次启动会打开 Ubuntu 默认的主页，如图 12-86 所示。



图 12-86 Firefox 初始化主页

Firefox 会在关闭的时候询问用户是否保存当前打开的页面标签，若用户选择“是”选项，

那么 Firefox 启动后会加载上一次访问所保存的网页；若用户选择“否”选项，那么 Firefox 下次启动会加载用户设置的主页。

2. 添加主页

使用 Firefox，用户可指定 Firefox 启动时加载的网站或单击工具栏上的主页图标时加载的网页。设置 Firefox 主页的具体操作如下所示：

- (1) 选择“编辑”|“首选项”菜单命令，弹出“首选项”对话框。
- (2) 选择“主要”选项卡，在“启动”选项的“启动 Firefox 时”下拉列表中选择“显示我的主页”选项，在“地址”文件框中输入 www.google.com 网址，如图 12-87 所示。



图 12-87 设置主页

- (3) 设置完毕，单击“确定”按钮，完成设置，关闭“首选项”对话框。

补充说明

在启动选项卡的“启动 Firefox 时”下拉列表中，用户还可设置“显示空白页”和“显示上次打开的窗口和标签页”。获取主页地址的方法，除了从键盘输入外，还可单击“使用当前页面”按钮获取当前 Firefox 所显示的页面；或是单击“使用书签”按钮，选择已保存为书签的页面地址；或者单击“恢复默认设置”按钮，使用主页还原为默认的网址。

3. 添加附加组件

由于 Firefox 拥有众多的用户，并且有很多计算机技术人员从事着 Firefox 附加组件的开发，所以用户学会在 Firefox 中使用附加组件可更加方便地学习、工作和娱乐。以添加“Flash Video Resources Downloader”组件为例，说明添加附加组件的具体步骤。操作步骤如下所示：

- (1) 选择“工具”|“附加组件”菜单命令，弹出“附加组件”对话框。
- (2) 选择“获取附加组件”选项卡，在搜索文本框中输入“flash”字符串，搜索有关 Flash 的附加组件，在搜索结果列表中选中“Flash Video Resources Downloader”选项。如图 12-88 所示。

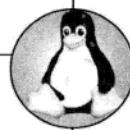


图 12-88 添加附加组件

(3) 单击“添加至 Firefox”按钮，执行添加附加组件的操作。

(4) Firefox 下载附加组件，并安装此组件，安装完毕之后，Firefox 提示用户重启 Firefox 应用程序。

(5) 重启 Firefox 后，可发现在 Firefox 的任务栏添加了一个新的图标。单击此图标就可从当前页面上获取 Flash 的视频。

补充说明

在“附加组件”对话框中，选择“插件”标签页，就可查看 Firefox 当前添加的插件。选择“扩展”标签页，就可查看 Firefox 已添加了的附加组件。

4. 添加书签

在 Firefox 中使用书签，可保存用户所需要的网页，为网页添加书签后，下次需要打开此页面时，单击相应的书签即可。添加书签的具体操作如下所示：

(1) 显示需要添加为书签的页面，然后选择“书签”|“将此页加为书签”菜单命令，弹出“编辑此标签”对话框。

(2) 在“标签”文本框中输入“影视”字符串，如图 12-89 所示。



图 12-89 编辑此书签

(3) 单击“完成”按钮即可。

补充说明

在“文件夹”下拉列表中，用户可设置书签的保存位置。标签是用来对书签进行分类的，这样用户就可查找到相同类别的页面。

5. 使用帮助

对于初学者而言，使用 Firefox 的帮助信息是学习 Firefox 的一个很好的方法。只要我们安装的时候完整安装了中文的 Ubuntu，那么此时 Firefox 也完美地中文化了，也就是说，Firefox 的帮助是中文的。选择“帮助”|“帮助内容”菜单命令，打开“Firefox 帮助”页面，如图 12-90 所示。



图 12-90 Firefox 帮助页面

在帮助页面中，用户可单击页面上显示的教程进行有选择的学习。用户还可在搜索工具栏文本框中输入关键字或词，搜索自己需要了解的内容。在搜索栏中，输入 cookie 字符串，单击搜索图标 \square ，搜索结果如图 12-91 所示。



图 12-91 搜索结果



6. 使用快捷键

Firefox 提供了很多快捷键，了解快捷键能使用户更加方便地操作 Firefox，常用快捷键如表 12-1 所示。

表 12-1 Firefox 浏览器快捷键

快 捷 键	说 明
Ctrl+N	新建 Firefox 窗口
Ctrl+T	在当前浏览器窗口中新建标签页
Ctrl+O	打开文件
Ctrl+Tab	切换到下一标签页
Ctrl+Shift+Tab	切换到上一标签页
Ctrl+W	关闭当前标签页
Ctrl+Shift+W	关闭当前 Firefox 窗口
Ctrl+F	查找
F5	重新载入
Alt+左方向键	后退
Alt+右方向键	前进
Ctrl+H	查看历史记录
Ctrl+S	网页文件另存为

12.7 虚拟机软件



虚拟化是一个广义的术语。在计算机方面，通常是指计算软件在虚拟的基础上而不是真实的基础上运行。虚拟化技术可以扩大硬件的容量，简化软件的重新配置过程。CPU 的虚拟化技术可以单 CPU 模拟多 CPU 并行，允许一个平台同时运行多个操作系统，并且应用程序都可以在相互独立的空间内运行而互不影响，从而显著提高计算机的工作效率。

Ubuntu 大约是自 Ubuntu 7.04 版本开始引入了一种全新的虚拟化技术，叫做 KVM，KVM 是 Kernel-based Virtual Machine 的缩写，其实和 Ubuntu 不相干，这是 Linux 内核级支持的虚拟化技术，只是恰好 Ubuntu 7.04 所用的 Linux 内核——Linux2.6.20 正好支持 KVM。Ubuntu 还支持众所周知的桌面虚拟机 VirtualBox。本节将介绍 KVM 和 VirtualBox 在 Ubuntu 中的使用。

12.7.1 内核全虚拟化 KVM

KVM 是指基于 Linux 内核的虚拟机，其全称是 Kernel-based Virtual Machine，其要求 CPU 支持虚拟化技术，在宿机上运行的速度快，能够很好地支持 SMP（对称多处理）。在 KVM 模型中，每一个虚拟机都是一个由 Linux 调度程序管理的标准进程；一个普通的 Linux 进程有两种运行模式：内核和用户。KVM 增加了第三种模式：客户模式——拥有自己的内核和用户模式。Ubuntu 9.04 所使用的内核是 Linux 2.6.28 的版本，自然也支持 KVM。

1. 检查 CPU 是否支持虚拟化

KVM 的使用要求计算机的 CPU 支持虚拟化，在 Ubuntu 中可使用如下命令检查 CPU 是否支持虚拟化：

```
$egrep '(vmx|svm)' /proc/cpuinfo
```

输入命令后，若出现结果，表示此计算机的 CPU 支持虚拟化；否则表示此计算机的 CPU 不支持虚拟化。当 CPU 支持虚拟化的时候，KVM 才可使用，以下的操作都是假设当前使用的计算机的 CPU 支持虚拟化。

2. 安装 KVM 和 Qemu

使用新立得软件包管理器安装 KVM 和 Qemu 软件包的方法如下所示：

- (1) 搜索 KVM，在搜索结果中为 KVM 软件包做下安装的标记。
- (2) 搜索 Qemu，在搜索结果中为 Qemu 软件包做下安装的标记。
- (3) 搜索 libvirt-bin，在搜索结果中为 libvirt-bin 软件包做下安装的标记。
- (4) 在新立得软件包管理器中应用所做的操作，安装已标记的软件包。

(5) 打开终端，输入命令：modprobe kvm-intel，若 CPU 不是 Intel，而是 ADM 的话，可使用命令：modprobe kvm-adm。

- (6) 安装 KVM 的桥接网络工具。在终端上，输入命令：sudo apt-get install bridge-utils。
- (7) 安装创建 TAP 界面的工具，在终端上输入命令：sudo apt-get install uml-utilities。
- (8) 配置桥接网络，使用 Gedit 文本编辑器打开/etc/network/interfaces 文件，将文件修改为：

```
auto eth0
iface eth0 inet static      //这里是 DHCP 设置所得
address 192.168.1.2
netmask 255.255.255.0
gateway 192.168.1.1

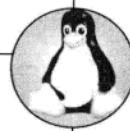
auto tap0
iface tap0 inet manual
up ifconfig $IFACE 0.0.0.0 up
down ifconfig $IFACE down
tunctl_user dranyu      //dranyu 是用户名

auto br0
iface br0 inet static
bridge_ports eth0 tap0
address 192.168.1.3
netmask 255.255.255.0
network 192.168.1.1
broadcast 192.168.1.255
gateway 192.168.1.1
```

- (9) 编辑文件之后，保存 interfaces 文件。激活 tap0 和 br0，输入如下命令：

```
sudo /sbin/ifup tap0
sudo /sbin/ifup br0
```

- (10) 完成 KVM 环境的配置。



补充说明

在配置桥接网络时，若 tap0 和 br0 没有如愿被激活，重启计算机即可。在终端使用 ifconfig 命令，可查看到当前计算机所用的网络链接接口，从运行结果中可发现新增了 tap0 和 br0 网络链接。

3. 安装 Windows XP 系统

使用 KVM 和 Qemu 可在 Ubuntu 宿主机上安装其他操作系统到由 KVM 创建的客户机上。以安装 windows XP 系统为例，讲述 KVM 的使用，具体的操作如下所示：

(1) 在主文件夹中创建一个名为 KVM 的文件夹，用来放置虚拟机的虚拟磁盘文件，使用命令如下所示：

```
mkdir KVM
```

(2) 创建虚拟磁盘文件。使用 CD KVM 命令进入 KVM 文件夹，然后输入如下命令创建虚拟磁盘文件 winXP.img：

```
qemu-img create -f vmdk winXP.img 10G
```

(3) 创建虚拟机。执行如下命令：

```
kvm -M pc -m 1024 -soundhw ac97 -hda /home/dranyu/kvm/winXP.img -localtime -net nic,vlan=0 -net user,vlan=0 -no-acpi -cdrom /home/dranyu/winXP.iso -boot d
```

上述命令解说：

-m 1024 表示为虚拟机分配 1024M 的内存空间。

-hda /home/dranyu/kvm/winXP.img 建立的虚拟磁盘。

-cdrom /home/dranyu/winXP.iso 指定安装光盘的路径。

-boot d 表示从光驱启动。

(4) 执行命令后，KVM 引导安装 WindowsXP 系统，它的安装过程如同在真机上安装的过程一样，这里就不多描述了。

补充说明

Qemu 是一套由 Fabrice Bellard 所编写的模拟处理器的自由软件。它拥有高速度及跨平台的特性，能模拟至接近真实电脑的速度。正是由于 Qemu 可模拟接近真实电脑的速度，KVM 将 Qemu 作为 I/O 硬件模拟器。在使用 qemu-img 命令时，使用了-f vmdk 的选项，它表示创建一个 vmdk 格式的磁盘。vmdk 格式是会自动增长的 vmware 类型的虚拟磁盘文件。

12.7.2 桌面虚拟机 VirtualBox OSE

Sun xVM VirtualBox 是由德国 InnoTek 软件公司出品的虚拟机器软件，现在则由升阳电脑公司进行开发，是升阳电脑 Sun xVM 虚拟化平台技术的一部分。它提供使用者在 32 位元或 64 位元的 Windows、Solaris 及 Linux 操作系统上虚拟其他 x86 的操作系统。使用者可以在 VirtualBox 上安装并且执行 Solaris、Windows、DOS、Linux、OS/2 Warp、OpenBSD 及 FreeBSD 等系统。2007 年 1 月，InnoTek 以 GNU 通用公共许可证（GPL）释出 VirtualBox 而成为自由软件，并提供二进制版本及开放源代码版本的程式码。这样 Ubuntu 也在发布时将 VirtualBox 放在它的软件源里。

VirtualBox OSE 是 Open Source Edition of VirtualBox 的简写，它的内核是 VirtualBox 内核的源码。它能够安装多个客户端操作系统，每个客户端系统皆可独立开启、暂停与停止。主

端操作系统与客户端操作系统皆能相互通信，多个操作系统同时运行的环境下，也能够彼此同时使用网络。

1. VirtualBox OSE 的安装和启动

在 Ubuntu 9.04 中使用 VirtualBox OSE，必须先安装 VirtualBox OSE 所涉及的软件包。使用新立得软件包管理器安装 VirtualBox OSE 的方法为：打开新立得软件包管理器，搜索“VirtualBox”字符串，在搜索结果中将 VirtualBox-Ose 标记为所需安装的软件包，然后应用更改，安装 VirtualBox 软件。

安装完毕后，选择 Ubuntu 顶部面板上“应用程序”|“附件”|“VirtualBox OSE”菜单命令，启动 VirtualBox OSE 应用程序，它的主界面如图 12-92 所示。



图 12-92 VirtualBox OSE 主界面

2. 创建 Xubuntu 的虚拟机

VirtualBox OSE 创建客户机的操作很简单，以创建 Xubuntu 的虚拟机为例讲述这一过程，它的具体操作步骤如下所示：

(1) 单击工具栏上的“新建”按钮，弹出“新建虚拟电脑”对话框，进入了引导用户新建虚拟电脑的第一步“欢迎使用新建虚拟电脑”，单击“下一步”按钮，进入“虚拟电脑名称和系统类型”步骤。

(2) 在“名称”文本框中输入“Xubuntu”，从“Operating System”下拉列表中选择“Linux”选项，在 version 的下拉列表中选择“Ubuntu”选项，如图 12-93 所示，单击“下一步”按钮，进入“内存”设置步骤。

(3) 移动滑块或在“内存大小”文本框中输入 512，如图 12-94 所示，单击“下一步”按钮，进入设置虚拟磁盘的步骤。



图 12-93 设置所要安装系统的类型

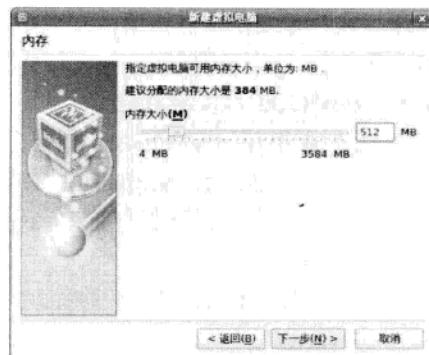


图 12-94 设置内存大小

(4) 由于我们是第一次创建虚拟电脑，还没创建虚拟硬盘。在这一步中，我们需要新建一个虚拟硬盘。单击“新建”按钮，开始“创建新的虚拟磁盘”的引导程序，在设置虚拟磁盘的参数时都保持默认值。创建虚拟磁盘后，如图 12-95 所示，单击“下一步”按钮，进入“摘要”步骤。

(5) 在摘要的信息中，显示了即将创建的虚拟电脑的配置，如图 12-96 所示，单击“完成”按钮，完成新虚拟电脑 Xubuntu 的创建。



图 12-95 设置虚拟硬盘

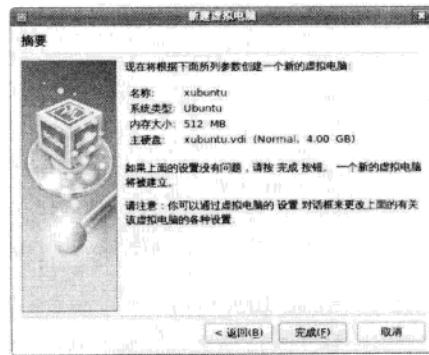


图 12-96 新建虚拟电脑的摘要

(6) 新的虚拟电脑创建之后，显示在 VirtualBox OSE 的主界面上，如图 12-97 所示。

3. 设置映像并安装 Xubuntu 系统

通过上述的步骤，我们已创建了一台虚拟电脑，可是我们并没为虚拟电脑添加一个光驱。创建虚拟电脑后，用户可单击“设置”按钮修改 Xubuntu 的配置。在安装 Xubuntu 系统前，先为 Xubuntu 客户机添加光驱，并设置加载 Xubuntu 的映像文件 Xubuntu-9.04-alternate-i386.iso，具体的操作步骤如



图 12-97 Xubuntu 虚拟机

下所示：

(1) 单击“设置”按钮，弹出“Xubuntu-设置”对话框，在左侧栏中选择“光驱”按钮，然后在光驱选项框中勾选“分配光驱”复选框，再选择“虚拟光盘”选项。

(2) 由于第一次使用虚拟光盘，我们要先单击下拉列表框后的调用虚拟光盘图标，弹出“Virtual Media Manager”对话框，单击“注册”按钮，然后获取已保存在磁盘中的 Xubuntu 映像文件 Xubuntu-9.04-alternate-i386.iso，添加虚拟光盘后，“Virtual Media Manager”对话框中显示如图 12-98 所示。单击“选择”按钮，完成虚拟光盘的选择，返回“Xubuntu-设置”对话框，如图 12-99 所示。



图 12-98 Virtual Media Manager 添加虚拟光盘

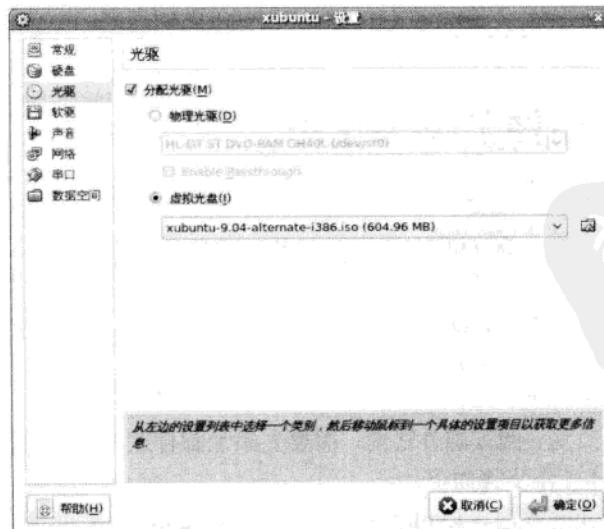


图 12-99 选择虚拟光盘



(3) 单击“确定”按钮完成虚拟光盘的设置。

添加 Xubuntu 的映像文件后，就可进行 Xubuntu 的安装了。安装 Xubuntu 的具体操作步骤如下所示：

(1) Xubuntu 的虚拟计算机配置好之后，Xubuntu 的安装就变得很简单了。在主界面中选择“Xubuntu”虚拟计算机，单击工具栏中的“开始”按钮，启动 Xubuntu 虚拟计算机。

(2) 启动后，出现如图 12-100 所示的画面。这表示 VirtualBox OSE 可读取光盘，接下来的操作步骤与在磁盘上通过光盘启动的安装方式一样，在此就不详细讲述了。

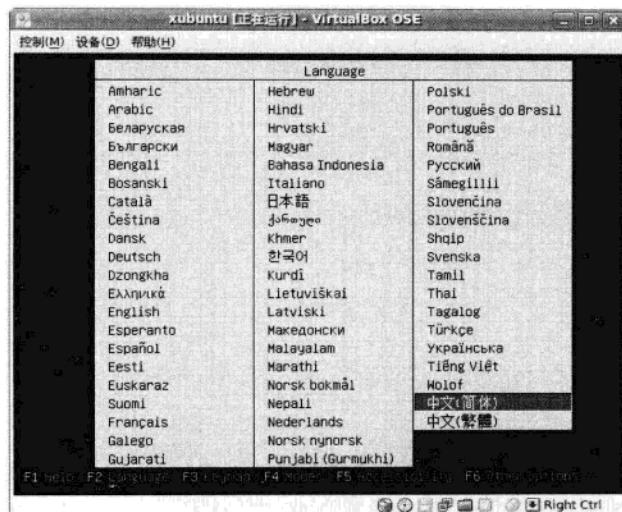


图 12-100 安装 Xubuntu

12.8 电子邮件 Evolution

由于各种原因，用户往往会拥有多个邮箱。邮件多了，多个邮箱操作会非常麻烦。为解决这一问题，Ubuntu 提供了 Evolution 应用程序可使用户方便快捷地收发和管理邮件。

Evolution 是一套组件，它集合邮件、行事历、地址簿与工作调度等功能于一身，并已默认安装到 Ubuntu 中。选择 Ubuntu 顶部面板的“应用程序”|“办公”|“Evolution 邮件及日历”菜单命令，即可启动 Evolution 应用程序。

1. 设置集成 Gmail 账号

在使用 Evolution 发送、阅读邮件之前，需要先添加邮件账号。在此以添加 Gmail 邮箱为例，添加和设置的具体操作步骤如下：

(1) 首次启动 Evolution 时，Evolution 将执行“Evolution 设置助手”应用程序，引导用户设置邮件账号。启动后，弹出“Evolution 设置助手”对话框，单击“前进”按钮。进入“从备份中恢复”步骤，没有 Evolution 的备份，单击“前进”按钮，进入“标识”的设置步骤，



在“电子邮件地址”文本框中输入 dranyu@gmail.com 字符串，其他可选信息可不填入，如图 12-101 所示。



图 12-101 输入邮箱地址

(2) 设置好“标识”后，单击“前进”按钮，进入“接收邮件”步骤，从“服务器类型”下拉列表中选择“POP”选项，在服务器中输入 Gmail 的 POP 服务器“pop.gmail.com”，在“使用安全链接”下拉列表中选择“SSL 加密”选项，勾选“记住密码”复选框，其他参数可不设置，如图 12-102 所示。

(3) 设置好“接收邮件”后，单击“前进”按钮，进入“接收选项”步骤，选择“Leave messages on server”选项，如图 12-103 所示。



图 12-102 接收电子邮件的设置



图 12-103 接收选项的设置

(4) 设置好“接收选项”后，单击“前进”按钮，进入“发送电子邮件”步骤，在“服务器类型”的下拉列表中选择“SMTP”选项；在服务器文本框中输入 Gmail 的 smtp 服务器“smtp.gmail.com”，在“使用安全链接”的下拉列表中选择“SSL 加密”选项；其他参数保持默认设置，如图 12-104 所示。

(5) 设置好“发送电子邮件”后，单击“前进”按钮，进入“时区”步骤，保持默认设置，单击“前进”按钮，进入“完成”步骤，如图 12-105 所示。



图 12-104 发送电子邮件的设置

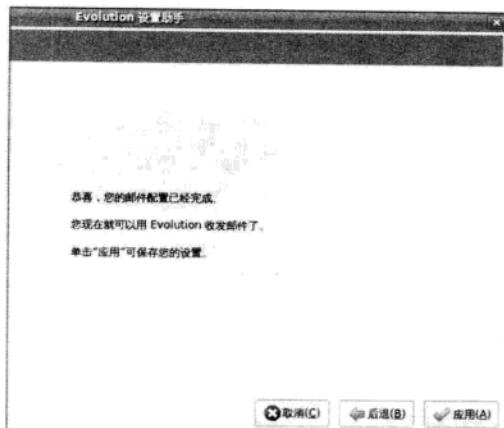


图 12-105 完成设置

(6) 单击“应用”按钮，完成添加 Gmail 账号的操作。完成设置后进入 Evolution 的主界面，如图 12-106 所示。

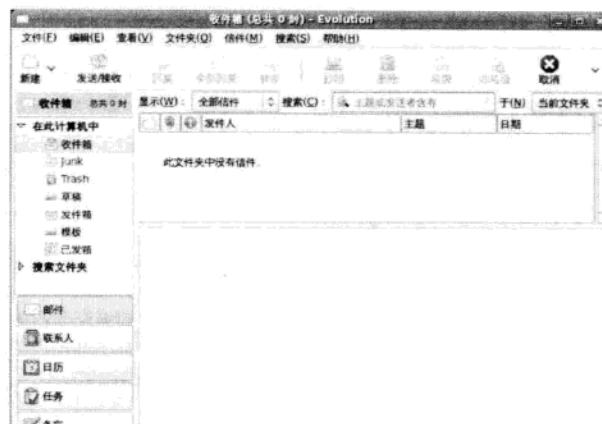


图 12-106 Evolution 的邮件主界面

补充说明

POP3 (Post Office Protocol 3) 即邮局协议的第 3 个版本，它规定怎样将个人计算机连接到 Internet 的邮件服务器和下载电子邮件的电子协议。它是因特网电子邮件的第一个离线协议标准，POP3 允许用户从服务器上把邮件存储到本地主机上，同时删除保存在邮件服务器上的邮件，而 POP3 服务器则是遵循 POP3 协议的接收邮件服务器，用来接收电子邮件的。SMTP (Simple Mail Transfer Protocol) 即简单邮件传输协议，它是一组用于由源地址到目的地址传递邮件的规则，由它来控制信件的中转方式。SMTP 协议属于 TCP / IP 协议族，它帮助每台计算机在发送或中转信件时找到下一个目的地。用户可通过选择“编辑” | “首选项”菜单命令，在“Evolution 首选项”对话框中的“邮件账号”修改邮件账号的设置。

2. 发送邮件

添加了 Gmail 的邮箱账号后，我们就可使用 Evolution 向网络发送电子邮件了。编辑及发送邮件的具体操作步骤如下所示：

- (1) 单击 Evolution 主界面左边的邮件按钮，切换到邮件功能界面，然后单击“新建”按钮，从下拉菜单中选择“信件”菜单命令，弹出“撰写新信件”窗口，如图 12-107 所示，在此用户可编辑新的信件。

- (2) 在“发件人”下拉列表中，可选择发件人的邮箱地址。此地址为已添加到 Evolution 的账号邮箱地址。上述我们已添加了一个邮箱地址，此处默认为已添加的默认邮箱地址。

- (3) 在“收信人”文本框中输入收件人的邮箱地址，如果有多个收件人，则每个收件人的邮件地址用逗号分开。如果 Evolution 已设置了“联系人”列表，则可以单击“收信人”按钮，从弹出的“从地址簿选择联系人”对话框中获取联系人的邮箱地址。

- (4) 在“主题”文本框中输入此电子邮件的主题。主题与文章的标题类似，方便收件人了解邮件的主要内容，当然不输入主题也是可以发送邮件的。



图 12-107 撰写新信件

- (5) 在填写完“收件人”和“主题”后，在邮件编号窗口最下方的文本框中输入邮件的正文。在“主题”文本框下方有“格式”工具栏，用户可以使用工具栏上的“格式”工具按钮来设置邮件正文的字体格式和段落格式。输入邮件的正文，如图 12-108 所示。



图 12-108 输入邮件正文



(6) 若需要向收件人发送文件、图片或其他文件，用户可使用附件功能。单击快捷工具栏上的“附件”按钮，从弹出的“插入附件”窗口中选择所需发送的文件或图片。插入附件，如图 12-109 所示。



图 12-109 插入附件

(7) 完成新信件的编辑后，单击工具栏上的“发送”按钮，邮件就会被发送出去。

3. 收取邮件

设置了 Evolution 的邮件地址账号，收取邮件的操作就很简单了。单击 Evolution 主界面左侧的“邮件”按钮，切换至邮件主界面，然后单击快捷工具栏上的“发送/接收”按钮即可发送和接收 Evolution 上账号的网络邮件，弹出“发送和接收邮件”对话框，如图 12-110 所示。

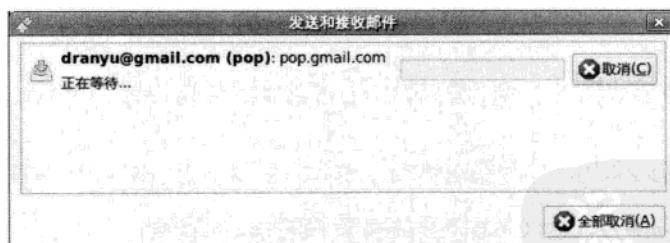


图 12-110 发送和接收邮件

单击“取消”按钮即可取消邮件的接收。若 Evolution 中设置了多个邮箱地址，那么此处会显示多个账号的接收邮件进度条。邮件接收完毕后，邮件会保存在 Evolution 的收件箱中，如图 12-111 所示。



图 12-111 收件箱中的邮件

4. 联系人管理

Evolution 应用程序集成了地址簿的功能。用户可使用地址簿记录生活中相关的人员信息，包括：联系人的电话，家庭地址，电子邮件地址，以及联系电话等，可方便地查找和管理联系人的信息。

启动 Evolution 后，单击主界面左侧的“联系人”按钮，切换至联系人主界面，如图 12-112 所示。

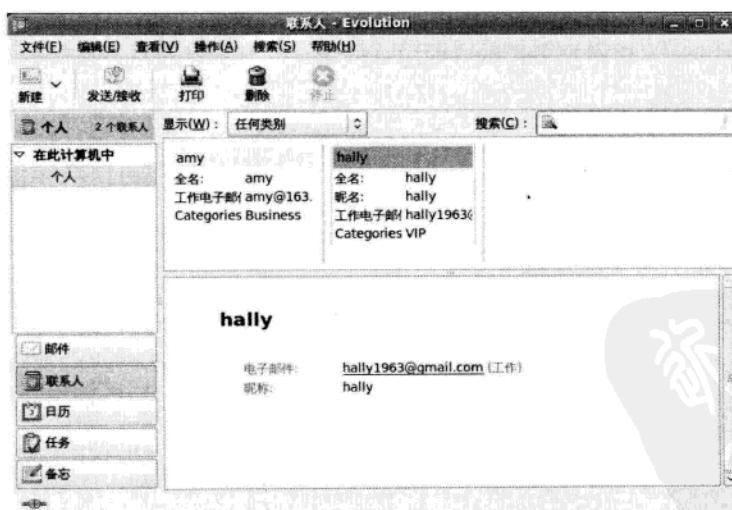


图 12-112 联系人主界面

在联系人主界面中，用户可从“显示”下拉列表中选择不同的显示方式，在搜索文本框



中输入字符串可搜索已添加到联系人地址簿上的联系人。

在 Evolution 中添加联系人的具体操作如下所示：

- (1) 当用户需要添加联系人时，在空白处双击，或者单击“新建”按钮从下拉菜单中选择“联系人”菜单，弹出“联系人编辑器”对话框，如图 12-113 所示。



图 12-113 “联系人编辑器”对话框

- (2) 在“联系人”标签页中输入 amy 的电子邮件和电话信息；在“个人信息”标签页中输入 amy 的个人信息，如个人主页、公司名称及其职位等；在“邮政地址”标签页上输入个人的家庭、工作或其他的邮政地址。

(3) 填写完毕后，单击左下角的“确定”按钮，保存新建联系人的信息。

补充说明

在 Evolution 中，添加联系人的方式有很多种。这里我们只介绍其中的一种，也是操作最简单的一种。Evolution 还提供了将其他邮箱的联系人导入为 Evolution 联系人的功能，它可导入 csv 格式的联系人文档。因而，使用 Evolution 可支持从 Outlook、Gmail，以及 163 等邮件系统的联系人信息的导入。

5. 设置日程事件

在日常生活中，事情繁多，若是没有合理地安排日程，事情会变得复杂而不可为。Evolution 提供了日程管理的功能，可使用户记录事情，并设置和安排日程，还可设置重要事情的提醒等。

启动 Evolution 后，单击 Evolution 主界面左侧的“日历”按钮，切换至日历主界面，如图 12-114 所示。



图 12-114 Evolution 的日历主界面

在日历中，设置约会提醒的具体操作步骤如下所示：

(1) 在日历主界面的左侧日历中，选择行程的日期。

(2) 在中间的列表框中双击所需设置约会的时间段的灰色行，弹出“约会-没有概要”窗口。在“概要”文本框中输入“踢球”字符串，在“位置”文本框中输入“老地方”，在“时间”栏中选择活动的时间段，在“描述”文本框中输入约会的注意问题或备注，如图 12-115 所示。

(3) 若用户需要为事件添加文本文件或图像，可使用其中的添加附件功能。单击工具栏上的“附件”按钮，弹出“添加附件”对话框，从磁盘上选择所要添加的附件，然后单击“打开”按钮完成附件的添加。添加附件后，如图 12-116 所示。

(4) 填写完毕后，单击工具栏上的“保存”按钮，完成约会事件的添加。

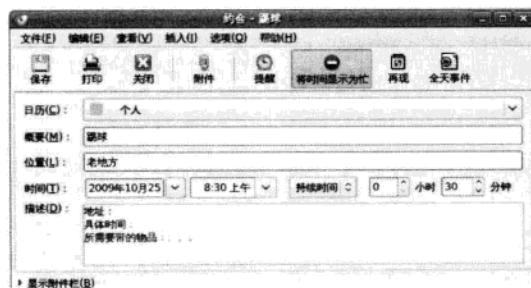
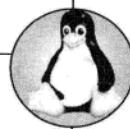


图 12-115 填写约会信息



图 12-116 添加附件



在约会对话框中，用户还设置约会的提醒，以及设置此约会的再现时间。

Evolution 是一款非常优秀的应用软件，其功能强大。Evolution 中还有我们日常使用的很多功能，在这里就不一一作介绍了，读者如在使用的过程中遇到难题，可在网上搜索相关资料进行学习。

12.9 小结



无论是文本编辑还是办公文档处理，无论是图片查看还是视频观看，Ubuntu 都能满足用户的需求。通过本章的学习，读者可掌握 Ubuntu 下应用软件的基本操作，可自行处理摄影图像，可使用电影播放器观看电影，应用即时通信工具与世界各地的朋友联系，方便使用 Firefox 在网络中遨游，在虚拟机环境中进行编程和测试，使用 Evolution 合理安排工作日程并与客户进行邮件交流。

12.10 习题



1. 使用 Gedit 文本编辑器打开 www.163.com 首页，然后将其高亮显示并保存到本地磁盘上。
2. OpenOffice.org 办公套件中包括哪几个应用程序？它们的主要功能是什么？
3. 在 Impress 中制作一张幻灯片，并为其添加切换动画。
4. 使用 F-Spot 导入主文件夹下的 pic 文件夹的图像。
5. 用鼠标右键单击保存在磁盘上的一张图像，选择 GIMP 应用打开此图像，并将此图像裁剪为 32×32 尺寸的图像。
6. 使用 aMule 的搜索功能，搜索“Ubuntu 视频教程”的视频，将视频保存在主文件夹下的 video 文件夹中，然后使用电影播放机 Totem 播放此视频文件。
7. 使用 Firefox 打开 www.163.com，并将此网址设置为 Firefox 的主页。
8. 为 Evolution 添加两个邮件账号，然后接收两个邮件账号里的邮件。
9. 将日常生活中的联系人信息输入到 Evolution 的联系人地址簿上。
10. 在 Evolution 中添加事件，并设置事件前半个小时提醒。

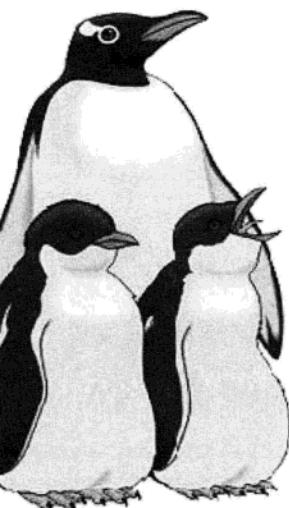
Linux

第5部分 高级应用

第13章

文件系统管理

文件系统是操作系统的重要组成部分，是用来存储和管理文件的子系统，而且每种操作系统所支持的文件系统数量和类型基本上各不相同。如 Windows 系统的操作系统 fat32、ntfs 等，而 Linux 除了支持 ntfs 还支持 ext2、ext3 等文件系统。熟悉 Linux 下的文件系统，可加深对 Linux 的理解，有助于更深入地掌握 Linux 的高级应用。本章主要介绍文件系统的概述、Linux 下常用的文件系统、磁盘分区工具的使用、磁盘配额，以及文件系统管理的基本操作命令。





13.1 文件系统概述

在早期的计算机系统中没有文件系统管理的机制，用户靠自行管理存储器上的信息。用户需要按照物理地址安排信息，组织数据输入输出，以致无法记住信息在存储介质上的分布情况。这种操作方式繁琐复杂、易出错、可靠性差。需求推动着存储器的发展，从而出现了大容量直接存储器。这一大容量直接存储器的问世为建立文件系统提供了良好的物质基础。接着，多道程序、分时系统的出现，以及多用户系统都能够同时共享大容量存储器，这促进了文件系统的发展。因而，对计算机系统中软件资源的管理形成了操作系统的文件系统。

文件系统作为操作系统最为重要的一部分，其定义了磁盘上存储文件的方法和数据结构。文件系统是操作系统组织、存取信息的重要手段，每种操作系统都有自己的文件系统。一个好的文件系统能提高操作系统的稳定性和可靠性，而不同的操作系统，对不同文件系统的支持是不一样的。

文件系统将一个文件分为多个文件块，然后以文件块的方式将信息存储在磁盘上，一个文件块通常会占用一个或多个磁盘扇区。磁盘上存取的最小单位是扇区，一个扇区是 512B，若遇到比较大的文件，一次只读取 512B 的数据，其效率非常低。因而，文件系统设定其读取数据的最小单位为块，也就是文件块。这样文件系统在读取一块数据的时候也就读取了多个扇区的数据，大大提高了数据的存取速度。

为了管理这些文件块，文件系统使用了文件分配表。文件分配表维护着文件在磁盘上的存放位置，例如需要读取和写入某个文件时，文件系统首先要到文件分配表中查出文件占用了多少个文件块，以及各个文件块的编号，然后根据文件块编号找到磁盘存储文件的物理位置，最后就可以根据要求进行读写操作。

对于 Linux 操作系统，其早期使用的是 Minix 文件系统。Minix 文件系统有较大的局限性，如文件名最长不能超过 14 个字符且最大文件大小为 64MB。后来在 1992 年 4 月出现了扩展文件系统 ext——专门为 Linux 设计的文件系统。到 1993 年，ext2 发布了，并添加到了 Linux 的内核中。目前，ext 2 也就成了 Linux 的标准文件系统。在 ext2 的基础上添加了系统日志记录之后就产生了 ext 3 文件系统。为了获取更强大的磁盘性能，Linux 内核自 2.6.28 开始正式支持新的文件系统 ext 4。当前 Ubuntu 9.04 已支持 ext 4 的文件系统。

在 Linux 的文件系统发展过程中，出现了一个比较重要的文件系统——虚拟文件系统 VFS。它将真实的文件系统从操作系统和文件系统中分离出来。它允许 Linux 支持多个不同的文件系统，每个文件系统都作为 VFS 的一个通用的软件接口。



13.2 Linux 支持的主要文件系统

虽然每种文件系统提供的功能与特点各不相同，但 Linux 可以支持多种文件系统。当然，

使用了 Linux 内核的 Ubuntu Linux 发行版本也支持多种文件系统，其中包括了 ext、ext2、ext3、ext4、reiserfs、vfs、iso9660、nfts 和 vfat 等。这就使得它更加灵活，可和许多操作系统共存。接下来，主要介绍几种 Ubuntu 支持的比较常用的文件系统。

13.2.1 ext2 文件系统

ext2 文件系统是 GNU/Linux 系统中的文件系统，由 Reu Card 设计的，其目标是为 Linux 提供一个强大的可扩展文件系统，同时也是 Linux 系统中设计最成功的文件。其的特点为存取文件的性能较好，它支持的最大容量可达到 16TB，文件名长度可达 255 字符。

ext2 文件系统是由逻辑块序列组成的，它把所使用的逻辑分区划分成块组，并从 0 开始依次编号。每个块组中包含若干数据块，数据块就是目录或文件的内容。块组中包含着几个用于管理和控制的信息块：超级块、组描述符表、块位图、inode 位图、inode 表和数据块。

- 超级块：它是记录整个文件系统相关信息的地方，它记录的信息主要有：
 - 块与 inode 的总量。
 - 未使用与已使用的 inode/块数量。
 - 一个块与一个 inode 的大小。
 - 文件系统的载入时间、最近一次写入数据的时间、最近一次检验磁盘的时间等文件系统的相关。
 - 有效位数值，若此文件系统已被载入，则有效位为 0；若未被载入，则有效位为 1。
- 组描述符表：记录此块由何处开始记录。
- 块位图：此处记录块是否使用。
- inode 位图：此处记录 inode 是否使用。
- inode 表：为每个 inode 的数据存放区。
- 数据块：为每个块的数据存放区。

Linux 默认情况下，使用的文件系统是 ext2，ext2 文件系统的确高效稳定。然而随着 Linux 系统在关键业务中的应用，Linux 文件系统 ext2 的弱点也渐渐暴露出来了。其中系统默认使用的 ext2 文件系统是非日志文件系统，这就是关键行业的一个致命弱点。

13.2.2 ext3 文件系统

ext3 文件系统是直接从 ext2 文件系统发展而来。目前，ext3 文件系统已经非常稳定可靠，它完全兼容 ext2 文件系统。用户可以平滑地从 ext2 文件系统中过渡到一个日志功能健全的文件系统。这实际上也是 ext3 日志文件系统设计的初衷。

ext3 拥有特点如下所示：

- 高可用性

系统使用了 ext3 文件系统后，即使在非正常关机后，系统也不需要检查文件系统。死机发生后，恢复 ext3 文件系统的时间只要数十秒钟。

- 数据的完整性

ext3 文件系统能够极大地提高文件系统的完整性，避免了意外关机对文件系统的破坏。在保证数据完整性方面，ext3 文件系统有 2 种模式可供选择。其中之一就是“同时保持文件



系统及数据的一致性”模式。采用这种方式，你永远不会再看到由于非正常关机而存储在磁盘上的垃圾文件。

- 文件系统的速度

尽管使用 ext3 文件系统时，有时在存储数据时可能要多次写数据，但是，从总体上来看，ext3 比 ext2 的性能还要好一些。这是因为 ext3 的日志功能对磁盘的驱动器读写头进行了优化。所以，文件系统的读写性能较之 ext2 文件系统来说，性能并没有降低。

- 数据转换

由 ext2 文件系统转换成 ext3 文件系统非常容易，只要简单地输入两条命令即可完成整个转换过程，用户不用花时间备份、恢复、格式化分区等。用一个 ext3 文件系统提供的小工具 tune2fs，它可以将 ext2 文件系统轻松转换为 ext3 日志文件系统。另外，ext3 文件系统可以不经任何更改，而直接加载成为 ext2 文件系统。

- 多种日志模式

ext3 有多种日志模式，一种工作模式是对所有的文件数据及 metadata（定义文件系统中数据的数据，即数据的数据）进行日志记录；另一种工作模式则是只对 metadata 记录日志，而不对数据进行日志记录，也即所谓 data=ordered 或者 data=writeback 模式。系统管理人员可以根据系统的实际工作要求，在系统的工作速度与文件数据的一致性之间作出选择。

13.2.3 ext4 文件系统

新一代的日志文件系统 ext4 是第 4 代的扩展文件系统，它在 ext3 的基础上作了很大的改进，而且与 ext3 是向后兼容的。目前，ext4 并不是一个 Linux 操作系统的标准，但它从竞争对手那里借鉴了许多有用的概念，并在 ext3 的基础上添加了新特点、新功能。ext4 文件系统的改进包括新特性、伸缩性和可靠性，当然也包括性能的改善，其中重要的几点改进如下所示：

- 与 ext3 向后和向前兼容性

ext4 与 ext3 是向前兼容的，这样就可以将 ext3 文件系统挂载为 ext4 文件系统。用户还可以将 ext4 挂载为 ext3（向后兼容），但前提是 ext4 文件系统不能使用区段。

- 提高了时间戳分辨率和扩展范围

ext4 之前扩展文件系统的时间戳都是以秒为单位的，而 ext4 设计时间戳时考虑到未来的发展，它将时间戳的单位提升到纳秒。ext4 给时间范围添加了两位，从而让时间寿命再延长 500 年。

- 增强了伸缩性

文件系统未来发展的一个重要方面就是伸缩性，即根据需求进行伸缩的能力。ext4 以多种方式实现了强大的伸缩性，它的伸缩性超越了 ext3，并且在文件系统源数据管理方面开辟了新领域。

- 突破了文件系统的限制

与 ext3 的一个明显差别就是 ext4 支持更大的文件系统、文件和子目录。ext4 支持的最大文件系统为 1EB，还支持最大 16TB 的文件。

- 提高了可靠性

ext4 文件系统可能会扩展得比较大，这将导致可靠性问题。但 ext4 通过许多自主保护和自主修复机制来解决这个问题，保障了系统的可靠性。

ext4 文件系统是高效、稳定、强大的文件系统，它随着文件系统研究的深入而不断发展，并且借鉴其他新文件系统的先进思想(比如 XFS、JFS、Reiser 和 IRON 容错文件系统技术)。

13.2.4 ReiserF 文件系统

ReiserFS 文件系统是由 Hans Reiser 和他领导的开发小组共同开发的，整个文件系统完全是从头设计的，是一个非常优秀的文件系统，也是最早用于 Linux 的日志文件系统之一。ReiserFS 的特点如下所示：

- 先进的日志机制

ReiserFS 有先进的日志 (Journaling/logging) 功能机制。日志机制保证了在每个实际数据修改之前，相应的日志已经写入硬盘。

- 高效的磁盘空间利用

ReiserFS 对一些小文件不分配 inode，而是将这些文件打包，存放在同一个磁盘分块中。而其他文件系统则为每个小文件分别放置到一个磁盘分块中。

- 独特的搜寻方式

ReiserFS 基于快速平衡树搜索，平衡树在性能上非常卓越，这是一种非常高效的算法。ReiserFS 搜索大量文件时，搜索速度要比 ext2 快得多。ReiserFS 在文件定位上速度非常快。在实际运用中，ReiserFS 在处理小于 4KB 的文件时，比 ext2 快 5 倍；带尾文件压缩功能的 ReiserFS 比 ext2 文件系统多存储 6% 的数据。

- 支持海量磁盘

ReiserFS 是一个非常优秀的文件系统，一直被用在高端 UNIX 系统上，可轻松管理上百 GB 的文件系统，ReiserFS 最大支持的文件系统为 16TB。

- 优异的性能

由于它的高效存储文件特点，使用 ReiserFS 文件系统的 PC，在启动 X 窗口系统时，所花的时间要比在同一台机器上使用 ext2 文件系统少 1/3。另外，ReiserFS 文件系统支持单个文件大小为 4GB 的文件，这为大型数据库系统在 Linux 上的应用提供了更好的选择。

13.2.5 VFS 文件系统

Virtual File System (或者被称为 Virtual Filesystem Switch，缩写为 VFS) 是 Linux 内核中的一个软件层，用于给用户空间的程序提供文件系统接口，屏蔽了文件系统的差别。在 VFS 管理下，Linux 不但能够读写各种不同的文件系统，而且还实现了这些文件系统相互之间的访问。

严格地说，VFS 并不是一个真正的文件系统。VFS 只存在于内存，不存在于任何外存空间，它在系统启动时建立，在系统关闭时消亡。它只作为一个连接真实文件系统和应用程序的接口而存在。在每次系统初始化期间，Linux 都会先在内存中构造一棵 VFS 的目录树，实际上便是在内存中建立相应的数据结构。VFS 目录树在 Linux 的文件系统模块中是个很重要的概念，不要将其与实际文件系统目录树混淆，VFS 中的各目录主要用途是用来提供实际文件系统的挂载点。



13.2.6 iso9660 文件系统

iso9660 是一个由国际标准化组织 (ISO) 为光盘媒介发布的文件系统。其目标是能够在不同的操作系统，如 Windows/Mac OS，以及类 UNIX 系统上交换数据。所有计算机平台将数据作为一个文件系统放在光盘上，文件系统被设计成为 UNIX、Linux、DOS、Windows 和 Mac 及它们的各种派生系统所公认。iso9660 要求以下几条限制：

- 目录树不可超过 8 级，即子目录名和子目录深度最多 8 级。
- 文件名加上扩展名必须少于 30 个字符，且目录不能使用扩展名。
- 文件名可以使用字母、数字和符号 (:@"&(*+,.-;/<=>?_)，但英文字母只能使用大写字母，不允许一些特殊字符如%或@。

13.2.7 其他文件系统

Ubuntu 中还支持其他的文件系统，如 nfs、vfat、ntfs 和 fat32 等文件系统。

1. Vfat 文件系统

Vfat 文件系统是伴随 Windows 95 推出而产生的，它对 fat16 文件系统进行扩展，支持 255 个字符的文件名，支持文件日期和时间属性，为每个文件保留了文件创建日期/时间、文件最近被修改的日期/时间和文件最近被打开的时期/时间。

2. Xfs 文件系统

Xfs 文件系统是 SGI 开发的高级日志文件系统，极具伸缩性，而且非常健壮。其主要特点包括以下几点：能够保证在宕机发生的情况下磁盘文件的数据完全性；采用优化算法实现快速查询与分配存储空间；能支持上百万 TB 的存储空间；能以接近裸设备 I/O 的性能存储数据。在单个文件系统的测试中，其吞吐量最高可达 7GB/秒，对单个文件的读写操作，其吞吐量可达 4GB/秒。

3. NFS 文件系统

网络文件系统 (NFS) 是一种在网络上的机器间共享文件的方法，是由 Sun 公司开发的，目前已经成为文件服务的一种标准 (RFC1904, RFC1813)。其最大的功能就是可以通过网络，让不同操作系统的计算机共享数据。NFS 文件服务器是 Linux 最常见网络的服务器之一。NFS 服务器可以看作是一个文件服务器，它可以让你的 PC 机通过网络将远端的 NFS 服务器共享的文件挂载到自己的系统中，在客户端使用 NFS 的远端文件就像是在使用本地文件一样。

13.3 Linux 文件系统目录结构



文件结构是文件存放在磁盘等存储设备上的组织方法，主要体现在对文件和目录的组织上。目录提供了管理文件的一个方便而有效的途径。文件系统是解决存储器如何存储文件数据的问题，而 Linux 文件系统目录结构是解决如何组织和管理文件的问题。

13.3.1 文件概念

在 Linux 系统下，任何软件和硬件设备都被看做是文件。我们都知道 Windows 的文件类型是很有必要的，若文件没有了文件后缀，我们就很难找到打开文件的相应应用程序。然而在 Linux 中文件可不带文件后缀。一个文件是否能被执行，和后缀名没有太大的关系，主要与文件的属性有关。可是在日常生活中，我们都会为 Linux 的文件添加上文件后缀，这是为了方便我们使用，而且添加上文件后缀之后，Linux 按照文件后缀调用相关应用程序打开文件。

Linux 文件类型和 Linux 文件的文件名所代表的意思是两个不同的概念。我们通过一般应用程序而创建的文件，如 file.txt、file.tar.gz，这些文件虽然要用不同的程序来打开，但放在 Linux 文件类型中衡量的话，大多是普通文件。

Linux 中常见的文件类型包括：普通文件、目录、字符设备文件、块设备文件、符号链接文件等，接下来详细讲述这几种文件类型。

1. 普通文件

其实，通常用户所接触到的文件多为普通文件，如图形文件、数据文件、文档文件、声音文件等都属于这种类型的文件。可使用 ls 命令查看文件类型，如例 13-1 所示。

例 13-1：查看主目录中文件的详细信息

查看用户主目录中文件的详细信息，执行命令如下所示：

```
dranyu@dranyu-desktop:~$ ls -la
总用量 484
drwxr-xr-x 70 dranyu dranyu 4096 2009-10-24 10:44 .
drwxr-xr-x  4 root   root   4096 2009-07-09 19:53 ..
-rw-----  1 dranyu dranyu 5588 2009-10-18 15:32 .bash_history
...
drw-r--r--  1 dranyu dranyu 3119 2009-08-24 10:27 doc
...
```

结果显示：在文件列表中有的文件记录行中包含“-rw-r--r--”的字符串，其中第一个符号是“-”，表示此文件在 Linux 中就是普通文件。

2. 目录

目录是用于存放文件名及其相关信息的文件，它是内核组织文件系统的基本节点。目录可包含下一级目录或其他文件。如在例 13-1 中，有的文件记录行包含了“drw-r--r--”字符串，第一个符号是“d”表示此文件是一个目录。

3. 设备文件

设备文件是用来表示 Linux 所支持的大多数设备。设备分为两种：字符设备和块设备。不同之处在于块设备对于请求有缓冲区。因此，它们可以选择以什么顺序进行响应。对于存储设备而言这一点是很重要的，因为在读写连续的扇区时比远远地分离的扇区更快。另一个不同就是块设备只能以块为单位接收输入和返回输出，而字符设备只能使用它们可能使用的或多或少的字节大小。大多数设备是字符设备，因为它们不需要这种缓冲而且不以固定块大小进行操作。

可以用 ls -l 区分一个设备文件是块设备还是字符设备。如果开始字符是“b”，它就是块设备；如果是“c”，就是字符设备，如例 13-2 所示。



例 13-2：查看/dev 目录下文件的详细信息

进入/dev 目录，然后执行如下命令查看文件的详细信息：

```
dranyu@dranyu-desktop:/dev$ ls -la
总用量 4
drwxr-xr-x 15 root root 3760 2009-10-23 21:21 .
...
brw-rw---- 1 root disk 8, 1 2009-10-23 21:21 sda1
crw-rw---- 1 root audio 14, 9 2009-10-23 21:19 dmmidi
lrwxrwxrwx 1 root root 13 2009-10-23 21:19 fd -> /proc/self/fd
...
```

从文件列表中可发现文件 sda1 记录行包含了“brw-rw----”字符串，而文件 dmmidi 记录行包含了“crw-rw----”字符串，表明 sda1 是块设备文件，dmmidi 是字符文件。

4. 符号链接文件

符号链接文件是一类特殊的文件，它包含有一条以绝对路径或者相对路径的形式指向其他文件或者目录的引用。根据链接文件的不同，它又可以细分为硬链接文件和软链接文件。硬链接文件其实也属于普通文件。而软链接文件不一样，它是另一种文件类型。从例 13-2 可发现，文件 fd 是一个符号链接文件（软链接文件），它是以“l”开头的。

补充说明

在 Linux 中创建一个文件时系统就为该文件分配一个唯一的 inode，文件的 inode 用于访问文件的属性。也就是说，Linux 系统中每个文件都要对应一个 inode，通俗点说 Linux 内核不认识字母，所以用户要有 UID、用户组要有 GID、进程要有 PID 等，文件要有 inode，一旦 inode 丢失文件将无法被访问。

13.3.2 目录结构

我们在安装 Ubuntu 的时候，可以将所有的 Linux 目录挂载到一个分区中，也可以为每个目录建立一个分区，而且分区的文件系统可不相同。熟悉 Linux 的目录组织形式，有利于用户组织和管理文件，以及维护 Linux 系统。Ubuntu 的文件系统目录结构是采用级层式的树状目录结构，在此结构中的最上层是根目录“/”，然后在其目录下包含其他子目录。在 Linux 中，虽说目录名称是可以自定义的，但是对于一些内核的文件目录名称已经成为一种约定，这些特殊的目录名称有着非常重要的功能，详细解说根目录“/”下的目录含义如表 13-1 所示。

表 13-1 Linux 系统默认目录

目 录	说 明
/	根目录是所有目录绝对路径的起始目录。如 home 目录的绝对路径为：/home
/bin	bin 是 binary 的缩写。这一目录存放了所有用户都可执行的且最经常使用的命令。如 cp、ls 和 mv 等
/boot	此目录是系统内核存放的目录，同时也是系统启动时所需文件的存放目录。如 vmlinuz 和 initrd.img。建议在安装 Ubuntu 时，为 boot 目录创建一个分区，有利于对系统进行备份
/dev	此目录保存了接口设备文件。如/dev/hda1、/dev/cdrom 等
/etc	此目录保存有关系统设置与管理的文件
/home	此目录放置所有普通用户的主文件夹或 FTP 站点目录。如 dranyu 的主文件夹为/home/dranyu

续表

目 录	说 明
/lib	此目录仅包含执行/bin 或/sbin 目录中的二进制文件时所需的共享函数库
/lost+found	在 ext2 或 ext3 文件系统中，当系统意外崩溃或机器意外关机，而产生一些文件碎片放在此目录。当系统启动的过程中 fsck 工具会自动检查此目录，并修复已经损坏的文件系统。有时系统发生问题，有很多的文件被移到这个目录中，可能会用手工的方式来修复，或将文件移到原来的位置上
/media	此目录主要为即插即用型存储设备的挂载点自动在此创建，比如 USB 盘系统自动挂载后，会在这个目录下产生一个目录；CDROM/DVD 自动挂载后，也会在这个目录中创建一个目录，类似 CDROM 的目录。这个只有在最新的发行套件上才有，比如 Fedora Core 4.0 5.0 等。可以参看/etc/fstab 的定义
/mnt	此目录主要是为各项设备的文件提供挂载点
/opt	此目录通常提供一个空间，以供较大型且固定的应用程序软件包保存文件之用，这可避免将文件分散到整个文件系统
/proc	此目录保存了当前系统内核与程序执行的信息，它和利用 ps 命令看到的内容相同
/root	此目录是 root 管理员的主文件夹
/sbin	sbin 是 System Binary 的缩写，此目录中存放系统启动时所需执行的程序，而且目录里的命令只有 root 才有权限执行
/tmp	tmp 是 Temporary 的缩写，此目录存放临时文件
/usr	此目录保存用户使用的系统命令，以及应用程序等信息。/usr 目录下包括涉及字体目录/usr/share/fonts，帮助目录 /usr/share/man 或/usr/share/doc，普通用户可执行文件目录/usr/bin 或/usr/local/bin，超级权限用户 root 的可执行命令存放目录，比如 /usr/sbin 或/usr/X11R6/sbin 或/usr/local/sbin 等；还有程序的头文件存放目录/usr/include
/var	var 是 Variable 的缩写，此目录的内容是经常变动的。/var 下有/var/log 这是用来存放系统日志的目录

补充说明

挂载点是一个目录，单击此目录可读取和查看它所挂载分区的内容。在 Linux 系统中，没有挂载的分区用户是访问不了保存在分区里的内容的。在第 4 章 Ubuntu 的安装过程中，我们就完成了系统默认的目录与磁盘分区的挂载关系。我们将 sda1 分区挂载到“/boot”目录，sda6 分区挂载到了“/”目录。在安装完毕之后，进入/boot 目录也就相当于查看 sda1 分区里的内容。在 Linux 中，用户必须为“/”根目录挂载一个分区，其他目录可不分配分区，Linux 会自动将放置到“/”目录下与“/”共享磁盘空间。



13.4 磁盘分区工具

在建立文件系统之前，首先应该保证对硬盘正确地分区。硬盘分区的信息存放在它的第一个扇区中。若使用的当前计算机只有一块硬盘，此扇区就是系统的主引导记录（MBR）。计算机引导启动时，BIOS 从该扇区读入引导信息并执行 MBR。Linux 下硬盘的分区类型包括三种：主分区、扩展分区和逻辑分区。Linux 规定主分区（或扩展分区）占用 1~16 个分区号的前 4 个。以 SATA 硬盘为例，主分区（或扩展分区）占用了 sda1、sda2、sda3 和 sda4；而逻辑分区占用 sda5~sda16 之间的 12 个分区号。其中扩展分区中一般都会被分解为逻辑分区。

主分区的作用就是用来启动计算机操作系统。因此，每个操作系统的启动程序或引导程



序都应该存放在主分区上。这就是主分区和扩展分区及逻辑分区的最大区别。所以，在安装 Ubuntu 的时候，我们将保存系统启动文件的目录 /boot 挂载分区 sda1，将根目录 / 挂载分区 sda6。

对于每个 Linux 分区，分区的大小和分区的类型是最主要的指标。容量的大小就是分区的容量，分区的类型规定了这个分区上文件系统的格式。Linux 通过分区类型号码来区分这些不同类型的分区。

Ubuntu 中提供了多种分区工具和磁盘管理工具，其中包括 fdisk、mkfs、mkswap、gparted 和 parted 等。在第 4 章中，我们已说过在 Live CD 系统上使用 gparted，但是 Ubuntu 系统并没有默认安装这一应用程序，用户可自行在新立得软件包管理器中安装 gparted。接下来讲述其他的分区工具 fdisk 和 parted。

补充说明

Linux 中有一种特殊的文件系统类型叫交换分区。这只是交换空间的一种形式。Linux 中的交换空间（Swap space）在物理内存被充满时被使用。如果系统需要更多的内存资源，而物理内存已经充满，内存中不活跃的页就会被移到交换空间去。虽然交换空间可以为带有少量内存的机器提供帮助。但是，这种方法不应该被当做是对内存的取代。交换空间位于硬盘驱动器上，它比进入物理内存要慢。交换空间可以是一个专用的交换分区，交换文件，或两者的组合。在我们安装 Ubuntu 之前就创建了一个大小约为内存大小 2 倍、文件系统类型为 swap 的交换分区。

13.4.1 fdisk 分区工具

fdisk 是各种 Linux 发行版本中最为常用的分区工具，是被定义为专家级别的分区工具，其实 fdisk 的基本用法并不复杂，初学者一样可学习并掌握它的使用。fdisk 工具能够完成分区的创建、删除、改变，以及显示等操作。

1. fdisk 的用法

在运行 fdisk 之前，用户必须获取到 root 的权限，然后在终端上输入不带参数的 fdisk 命令，那么 fdisk 的基本语法就会显示在终端上，如例 13-3 所示。

例 13-3：不带参数的 fdisk 命令

执行不带参数的 fdisk 命令如下所示：

```
root@dranyu-desktop:/home/dranyu# fdisk
Usage: fdisk [-l] [-b SSZ] [-u] device
E.g.: fdisk /dev/hda (for the first IDE disk)
      or: fdisk /dev/sdc (for the third SCSI disk)
      or: fdisk /dev/eda (for the first PS/2 ESDI drive)
      or: fdisk /dev/rd/c0d0 or: fdisk /dev/ida/c0d0 (for RAID devices)
```

执行结果中第一行 usage 显示了 fdisk 的语法结构。在“E.g.”后面的是例子。使用 fdisk 时需要注意，若 fdisk 命令含有参数，则参数列表中一定要包含设备参数。

例 13-4：显示硬盘分区的信息

指定设备参数为 /dev/sda，查看 SATA 硬盘的使用情况，执行命令和结果显示如下所示：

```
root@dranyu-desktop:/home/dranyu# fdisk -l /dev/sda
Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000123cc

      Device Boot      Start        End      Blocks   Id  System
  /dev/sda1   *           1         136    1092388+   83  Linux
  /dev/sda2            137       1044     7293510     5  Extended
  /dev/sda5            137       270    1076323+   82  Linux swap / Solaris
  /dev/sda6            271       877    4875696     83  Linux
```

在结果显示中，第一行显示了当前磁盘空间的总大小是 8589MB；第二行显示了磁盘的相关信息，如磁盘的柱面数是 1044；第四行显示的是柱面的数量单位，即每个柱包括的字节数，其值是 8225280 字节；第五行表示磁盘的标识；后面的数据表格显示了各个分区的磁盘空间。

2. 进入磁盘分区编辑界面

fdisk 工具提供命令操作界面，操作界面中用户可使用相关的命令完成对磁盘的管理操作，如例 13-5 所示。

例 13-5：进入 fdisk 的命令操作界面

执行命令和结果显示如下所示：

```
root@dranyu-desktop:/home/dranyu# fdisk /dev/sda
```

```
The number of cylinders for this disk is set to 1827.
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):
```

此时，显示结果中最后一行是 fdisk 的命令提示符。在命令提示符中，输入 m，然后按下车键，显示 fdisk 命令界面中各命令的使用说明，如例 13-6 所示。

例 13-6：显示 fdisk 命令界面的各命令

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

fdisk 的所有内部命令都显示出来了。其中常用选项的说明如下所示：n 命令表示创建一个新分区；d 命令表示删除一个已存在的分区；p 命令表示显示分区列表；t 命令表示修改分区的系统 ID 号；l 命令表示显示所支持的分区类型列表；a 命令表示指定启动分区；w 命令表示将对分区表的修改存盘并退出。

3. 建立分区

在使用文件系统之前，需要为文件系统创建一个磁盘分区，这就要求我们进行建立分区的操作。在 Linux 的分区过程中，首先需要使用 p 命令来显示磁盘分区表信息，然后根据信息确定将来的分区，具体的操作步骤如下所示：

(1) 输入 p 命令，查看分区表信息，执行结果如下所示：

```
Command (m for help): p
Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000123cc
      Device Boot      Start        End      Blocks   Id  System
  /dev/sda1   *           1         136    1092388+   83  Linux
```



/dev/sda2	137	1044	7293510	5	Extended
/dev/sda5	137	270	1076323+	82	Linux swap / Solaris
/dev/sda6	271	877	4875696	83	Linux

(2) 可发现/dev/sda1 是主分区, /dev/sda2 是扩展分区, /dev/sda5 和/dev/sda6 是逻辑分区。sda6 的结束扇区 end 是 877, 而 sda2 的结束扇区 (end) 是 1044, 这表示从 878 扇区到 1044 扇区属于未定义的磁盘空间。输入 n 命令, 开始进入创建新分区操作, 显示第一步, 选择新建分区的类型, 是主分区 (primary partition) 还是逻辑分区 (logical), 选择 1, 然后按下回车键。fdisk 提示输入开始的磁盘扇区号, 我们按照默认的扇区填写即可, 即输入 878, 然后按下回车键。fdisk 提示输入结果的磁盘扇区号或输入新建分区的大小值, 我们输入+500M, 然后按下回车键, 完成了新建分区的操作, 代码如下所示:

```
Command (m for help): n
Command action
  l  logical (5 or over)
  p  primary partition (1-4)
l
First cylinder (878-1044, default 878): 878
Last cylinder, +cylinders or +size{K,M,G} (878-1044, default 1044): +500M
```

(3) 使用 p 命令查看创建新分区后的分区表信息, 代码如下所示:

```
Command (m for help): p
Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000123cc
  Device Boot      Start        End     Blocks   Id  System
/dev/sda1  *          1       136    1092388+  83  Linux
/dev/sda2            137      1044    7293510   5  Extended
/dev/sda5            137      270    1076323+  82  Linux swap / Solaris
/dev/sda6            271      877    4875696   83  Linux
/dev/sda7            878      942    522081    83  Linux
```

可发现多了/dev/sda7 分区, 其大小为 522081, 这表示新建分区成功。

补充说明

fdisk 工具没有提供调整分区大小的功能, 若磁盘上没有未定义的磁盘空间, 那么使用 n 命令不可创建新的分区。也就是说, 当最后一个逻辑分区的结束扇区号与扩展分区的结束扇区号相同时, 磁盘上没有未定义的磁盘空间。其实使用 p 命令可知此磁盘的总扇区的大小是 1044, 显示在第二行的“255 heads, 63 sectors/track, 1044 cylinders”中。在创建新的分区的时候, 还需注意的是当磁盘中没有创建扩展分区时我们是不能创建逻辑分区的。那时, fdisk 会提示是创建扩展分区 (extended) 还是创建主分区 (primary partition), 选择“e”表示将创建扩展分区。

4. 修改文件系统类型 ID

Ubuntu 中为它所支持的文件系统类型设置了编号。我们所创建的新分区的分区文件系统类型默认是 Linux, ID 是 83。接下来我们来修改/dev/sda7 分区的文件系统类型, 将其改为 NTFS 类型的文件系统, 具体操作如下所示:

(1) 输入 t 命令, 输入 7 作为“Partition number”, 也就是我们要修改的分区号。

```
Command (m for help): t
Partition number (1-7): 7
```

(2) 输入所要执行修改操作的分区号之后会提示“Hex code(type L to list codes)”，输入 L，然后按下回车键，查看 Linux 下所支持的文件系统类型 ID。

```
Hex code (type L to list codes): L
```

0 Empty	1e Hidden W95 FAT1 80 Old Minix	bf Solaris
1 FAT12	24 NEC DOS	81 Minix / old Lin c1 DRDOS/sec (FAT-
2 XENIX root	39 Plan 9	82 Linux swap / So c4 DRDOS/sec (FAT-
3 XENIX usr	3c PartitionMagic	83 Linux
4 FAT16 <32M	40 Venix 80286	84 OS/2 hidden C: c7 Syrinx
5 Extended	41 PPC PReP Boot	85 Linux extended da Non-FS data
6 FAT16	42 SFS	86 NTFS volume set db CP/M / CTOS / .
7 HPFS/NTFS	4d QNX4.x	87 NTFS volume set de Dell Utility
8 AIX	4e QNX4.x 2nd part	88 Linux plaintext df BootIt
9 AIX bootable	4f QNX4.x 3rd part	8e Linux LVM
a OS/2 Boot Manag	50 OnTrack DM	93 Amoeba
b W95 FAT32	51 OnTrack DM6 Aux	94 Amoeba BBT
c W95 FAT32 (LBA)	52 CP/M	9f BSD/OS
e W95 FAT16 (LBA)	53 OnTrack DM6 Aux	a0 IBM Thinkpad hi ee GPT
f W95 Ext'd (LBA)	54 OnTrackDM6	a5 FreeBSD
10 OPUS	55 EZ-Drive	a6 OpenBSD
11 Hidden FAT12	56 Golden Bow	a7 NeXTSTEP
12 Compaq diagnost	5c Priam Edisk	a8 Darwin UFS
14 Hidden FAT16 < 3	61 SpeedStor	a9 NetBSD
16 Hidden FAT16	63 GNU HURD or Sys ab	Darwin boot fb VMware VMFS
17 Hidden HPFS/NTF	64 Novell Netware	b7 BSDI fs fc VMware VMKCORE
18 AST SmartSleep	65 Novell Netware	b8 BSDI swap fd Linux raid auto
1b Hidden W95 FAT3	70 DiskSecure Mult bb	Boot Wizard hid fe LANstep
1c Hidden W95 FAT3	75 PC/IX	be Solaris boot ff BBT

每两个字段表示一种文件系统类型，如“7 HPFS/NTFS”。7 表示文件系统的 ID，HPFS/NTFS 表示文件系统的类型。

(3) 从列表中找到 NTFS 文件系统类型的 ID 是 7，然后在提示符中输入 7，按下回车键，将/dev/sda7 的文件系统类型修改为 NTFS。

```
Hex code (type L to list codes): 7
Changed system type of partition 7 to 7 (HPFS/NTFS)
```

5. 删除分区

fdisk 没有调整分区的功能，可是我们可以使用删除分区和添加分区调整分区的大小。删除分区的操作十分简单，在 fdisk 命令界面的命令提示符上输入 d，然后选择所要删除的分区号即可。如删除/dev/sda7 分区，代码如下所示：

```
Command (m for help): d
Partition number (1-7): 7
```

输入所要删除的分区号后按下回车键完成操作。在删除分区时需十分谨慎，不要将有重要数据的分区删除了。

6. 应用更改并退出 fdisk

在 fdisk 命令操作界面中，若不希望 fdisk 执行对磁盘所做的操作，可输入 q 命令直接退出 fdisk 命令操作界面；若要求 fdisk 执行磁盘所做的操作，就一定要输入 w 命令应用对磁盘所做的修改操作。执行 w 命令，代码如下所示：



```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
```

执行完成 w 命令后，退出 fdisk 命令操作界面，返回到终端命令提示符状态。

13.4.2 parted 分区工具

parted 分区工具是 Linux 内置的一个分区调整工具，也同样可以做到 fdisk 类似功能，并且方便地调整分区的容量。其最大优点是允许在不损失硬盘中原有数据的前提下对硬盘进行重新设置分区、分区格式化，以及复制、移动、格式转换和更改磁盘分区大小、隐藏硬盘分区，以及多操作系统启动设置等操作。

注意

在使用 parted 分区工具时一定要注意，因为在 parted 中对分区所做的任何操作都会立即应用到磁盘中。parted 使用不当会给用户造成不必要的损失。

1. 启动 parted

启动 parted，可在终端输入不带参数的 parted 或输入带设备名参数的 parted 命令。若不带参数，parted 会使用当前的磁盘作为操作的对象，如例 13-7 所示。

例 13-7：不带参数的 parted 命令

```
root@dranyu-desktop:/home/dranyu# parted
GNU Parted 1.8.8
使用 /dev/sda
欢迎使用 GNU Parted! 输入 'help' 可获得命令列表
(parted)
```

此时进入了 parted 的命令操作界面，(parted) 是命令提示符。

2. 帮助说明

启动后，在命令提示符后输入 help，然后按回车键，查看 parted 的使用说明，如图 13-1 所示。

```
root@dranyu-desktop:/home/dranyu# parted
GNU Parted 1.8.8
使用 /dev/sda
帮助
  parted help          do a simple check on the file system
  check [NUMBER]        检查文件系统
  cp [IRON-DEVICE] FROM-MINOR TO-MINOR   将文件系统复制到另一个分区
  help [COMMAND]        print general help, or help on COMMAND
  mklabel [mktable] LABEL TYPE             创建一个新磁盘分区表
  mkfs NUMBER FS-TYPE  在分区创建新的文件系统 FS-TYPE
  mklabel [mktable]          创建新的分区表
  mkpartfs 分区类型 文件系统类型 从始点 纵止点    创建一个带有文件系统的分区
  move NUMBER 插入点 纵止点 移动卷号为NUMBER的分区
  name DEVICE[PART] NAME      将卷号为NUMBER的文件系统名为NAME
  print [devices]list all[NUMBER]  显示分区表，可用
  devices, free space, all found partitions, or a particular partition
  quit                  退出程序
  rescue [从始点] [纵止点] "从始点" 的逻辑分区
  resize NUMBER START END    调整分区NUMBER和它的文件
  system                 系统
  rm NUMBER              删除卷号为NUMBER的分区
  set NUMBER FLAG STATE    改变FILESYSTEM上分区NUMBER的FLAG
  toggle [NUMBER] [FLAG]    切换FILESYSTEM上分区NUMBER的FLAG
  unit UNIT               设置默认单位为UNIT
  version                显示版本号
  copyright information of GNU Parted
(parted)
```

图 13-1 help 帮助文档

其中 check 命令执行文件系统的简单检查；cp 命令对文件系统进行复制操作；mkfs 命令创建指定类型的文件系统；mkpart 命令不创建新文件系统而只是制作分区；move 命令移动分区；quit 命令退出 parted 命令操作界面；resize 命令调整分区的大小；rm 命令删除指定分区；print 命令显示分区表。

3. 创建文件系统

在 parted 中，创建新分区有两个步骤，第一是新建分区，第二是为新建分区指定文件系统类型。具体的操作步骤如下所示：

(1) 显示分区列表信息，在命令提示符输入 print 命令，代码如下所示：

```
(parted) print
磁盘 /dev/sda: 8590MB
Sector size (logical/physical): 512B/512B
分区表: msdos
数字 开始 大小 类型 文件系统 标志
1 32.3kB 1119MB 1119MB primary ext4 启动
2 1119MB 8587MB 7469MB extended
5 1119MB 2221MB 1102MB logical linux-swap
6 2221MB 7214MB 4993MB logical ext4
```

(2) 使用 mkpart 命令新建分区。执行代码如下所示：

```
(parted) mkpart logical ext4 7214 8000
```

执行后使用 print 命令查看新建结果。

```
(parted) print
磁盘 /dev/sda: 8590MB
Sector size (logical/physical): 512B/512B
分区表: msdos
数字 开始 大小 类型 文件系统 标志
1 32.3kB 1119MB 1119MB primary ext4 启动
...
7 7214MB 8000MB 786MB logical
```

(3) 为 7 分区设置文件系统。使用 mkfs 创建文件系统，代码如下所示：

```
(parted) mkfs
警告: The existing file system will be destroyed and all data on the partition will
be lost. Do you want to continue?
parted: invalid token: 7
是/Yes/否/No? Yes
分区编号? 7
文件系统类型? [ext2]? ext2
```

执行后使用 print 命令查看新建结果。

```
(parted) print
磁盘 /dev/sda: 8590MB
...
数字 开始 大小 类型 文件系统 标志
...
7 7214MB 8000MB 786MB logical ext2
```

(4) 到此，完成了新分区/dev/sda7 的创建操作。

补充说明

目前，parted 还不支持 ext4 文件系统的创建。在 parted 中，分区的开始使用数值是在整个磁盘中的磁盘空间大小，而不像 fdisk 那样使用扇区的数值。



4. 调整分区

parted 分区工具提供了 fdisk 不具有的功能——调整分区的大小。当分区的大小不符合我们的要求时，可使用 parted 中的 resize 命令方便地调整分区的大小。resize 命令的语法格式为“`resize 分区号 分区起始 分区结束`”。

例 13-8：调整分区大小

将分区号为 7 的分区大小调整为 500M，代码如下所示：

```
(parted) resize 7 7214 7714  
(parted) print  
...  
数字 开始:   End     大小      类型      文件系统      标志  
...  
7    7214MB  7714MB  500MB    logical    ext2
```



13.5 文件系统管理基本命令

在进行软件安装、资料下载的时候，检查磁盘的空间和了解磁盘的使用情况很有必要。Ubuntu 提供了多个磁盘管理命令，能够有效地使用和管理磁盘空间。接下来，我们主要介绍常用的磁盘空间管理命令的使用。

13.5.1 dd 命令——磁盘备份和数据转换

Ubuntu Linux 中 du 命令把指定的输入文件复制到指定的输出文件中，并且在复制过程中可进行格式转换，也就是说使用 dd 命令可以帮助我们将软盘中的数据写成一个文件，或将硬盘中的文件写入软盘等。

1. dd 的语法

dd 命令的语法为：`dd [选项]`

其中部分选项的说明如下所示：

```
if = 输入文件（或设备名称）  
of = 输出文件（或设备名称）  
ibs = bytes 一次读取 bytes 字节，即读入缓冲区的字节数  
skip = blocks 跳过读入缓冲区开头的 ibs*blocks 块  
obs = bytes 一次写入 bytes 字节，即写入缓冲区的字节数  
bs = bytes 同时设置读/写缓冲区的字节数（等于设置 ibs 和 obs）  
cbs = byte 一次转换 bytes 字节  
count=blocks 只复制输入的 blocks 块  
conv = ASCII 把 EBCDIC 码转换为 ASCII 码  
conv = ebcDIC 把 ASCII 码转换为 EBCDIC 码  
conv = ibm 把 ASCII 码转换为 alternate EBCDIC 码  
conv = block 把变动位转换成固定字符  
conv = ublock 把固定位转换成变动位  
conv = ucase 把字母由小写转换为大写  
conv = lcase 把字母由大写转换为小写  
conv = notrunc 不截断输出文件  
conv = swab 交换每一对输入字节  
conv = noerror 出错时不停止处理
```

`conv = sync` 把每个输入记录的大小都调到 `ibs` 的大小 (用 NUL 填充)

2. 操作实例

例 13-9: 将整盘备份到指定的磁盘分区上

若计算机有两个硬盘分区 `hdx` 和 `hdy`, 我们可使用 `dd` 命令将 `hdx` 上 1000KB 空间的数据备份到 `hdy` 中, 具体的操作命令如下所示:

```
# dd if=/dev/hdx of=/dev/hdy bs=1000k
```

`/dev/hdx` 就是输入的文件, 而 `/dev/hdy` 就是输出的文件, `dd` 将完成数据的备份制作。

例 13-10: 将磁盘数据备份到指定路径的 `image` 文件

有时我们并不需要将磁盘数据备份到其他的磁盘中, 并且以 1024KB 每次的速度读取磁盘上的数据, 以 2048KB 每次的速度写入映像文件, 那么我们还可将磁盘数据备份到指定的 `image` 映像文件中, 具体的制作命令如下所示:

```
# dd if=/dev/hdx of=~/image ibs=1024k obs=2048k
```

例 13-11: 将备份的 `image` 文件恢复到指定的磁盘分区中

在磁盘分区受到损坏时, 我们可使用 `dd` 命令恢复已做备份的磁盘, 具体制作命令如下所示:

```
# dd if=~/image of=/dev/hdx
```

补充说明

唯有 root 管理员拥有使用 `dd` 命令的权限, 在使用的时候, 一定要确定是否已获得 root 身份。

13.5.2 df 命令——检查文件系统占用空间情况

Ubuntu 中提供 `df` 命令检查文件系统的磁盘空间占用情况。通过 `df` 命令可以获取各文件系统在硬盘上占用了多少空间, 以及磁盘目前还剩下多少空间等信息。

1. df 的语法

`df` 命令的语法为: `df [选项]`

Ubuntu 中 `df` 命令部分选项的含义如下:

- `-a`: 显示所有文件系统的磁盘使用情况, 包括 0 块 (block) 的文件系统, 如 `/proc` 文件系统。
- `-k`: 以 k 字节为单位显示。
- `-i`: 显示 i 节点信息, 而不是磁盘块。
- `-t`: 显示各指定类型的文件系统的磁盘空间使用情况。
- `-x`: 列出不是某一指定类型文件系统的磁盘空间使用情况 (与 `t` 选项相反)。
- `-T`: 显示文件系统类型。

2. 操作实例

例 13-12: 以字节 k 为单位显示磁盘空间的占用情况

```
dranyu@dranyu-desktop:~$ df -k
文件系统      1K-块    已用    可用  已用%  挂载点
/dev/sda6        9835552   5084420   4251516  55%  /
tmpfs            559880       0   559880   0%  /lib/init/rw
```



```
varrun          559880      348    559532   1% /var/run  
.....  
/dev/sdal       3612272    124488  3304244   4% /boot
```

在输出结果中，第一列表示的是文件系统对应的设备文件的路径名，第 2 列表示的是相应分区所包含的数据块数目，第 3 列表示已被占用的数据块数目，而第 4 列表示相应的文件系统中可用的数据块数目，第 5 列表示磁盘空间的使用率，第 6 列表示文件系统的挂载点。

例 13-13：以 MB 或 GB 单位显示空间

目前，我们使用的磁盘空间都比较大，以 KB 显示空间并不能直观地查看空间的使用情况。此时，我们可使用-h 选项，以 MB 或 GB 单位显示空间的使用情况，代码如下所示：

```
dranyu@dranyu-desktop:/$ df -h  
文件系统 容量 已用 可用 已用% 挂载点  
/dev/sda6  9.4G 4.9G 4.1G 55% /  
tmpfs     547M 0 547M 0% /lib/init/rw  
.....  
/dev/sdal   3.5G 122M 3.2G 4% /boot
```

例 13-14：显示所有文件系统的磁盘使用情况

在一般情况下，df 命令不会显示隐藏的文件系统（如 sysfs、proc 等）的磁盘使用情况，使用-a 则可显示所有文件系统的磁盘使用情况，代码如下所示：

```
dranyu@dranyu-desktop:/$ df -a  
文件系统 1K-块 已用 可用 已用% 挂载点  
/dev/sda6 9835552 5084424 4251512 55% /  
proc      0 0 0 - /proc  
sysfs    0 0 0 - /sys  
.....  
gvfs-fuse-daemon 0 0 0 - /home/dranyu/.gvfs
```

例 13-15：显示文件系统的类型

若需要了解文件系统的类型，可在使用 df 时添加-T 选项，代码如下所示：

```
dranyu@dranyu-desktop:/$ df -Th  
文件系统 类型 容量 已用 可用 已用% 挂载点  
/dev/sda6 ext3 9.4G 4.9G 4.1G 55% /  
tmpfs   tmpfs 547M 0 547M 0% /lib/init/rw  
.....  
/dev/sdal ext3 3.5G 122M 3.2G 4% /boot
```

13.5.3 du 命令——显示文件占用磁盘空间情况

du 的英文原义为“disk usage”，含义为显示文件占用磁盘空间的使用情况。

1. du 的语法

语法：

```
du [选项] [Names...]
```

说明：Linux 中 du 命令逐级进入指定目录的每一个子目录并显示该目录占用文件系统数据块（1024 字节）的情况。若没有给出 Names，则对当前目录进行统计。

Ubuntu 中 du 命令的部分选项含义如下所示：

- -s：对每个 Names 参数只给出占用的数据块总数。
- -a：递归地显示指定目录中各文件及子孙目录中各文件占用的数据块数。若既不指定-s，也不指定-a，则只显示 Names 中的每一个目录及其中的各子目录所占的磁盘块数。

- **-b:** 以字节为单位列出磁盘空间使用情况（系统默认以 k 字节为单位）。
- **-k:** 以 1024 字节为单位列出磁盘空间使用情况。
- **-c:** 最后再加上一个总计（系统默认设置）。
- **-l:** 计算所有的文件大小，对硬链接文件，则计算多次。
- **-x:** 跳过在不同文件系统上的目录不予统计。

2. 操作实例

例 13-16: 不带参数的 du 命令

若使用了不带参数的 du 命令，du 命令显示当前目录下所有文件，以及其子目录里的磁盘空间使用情况。

```
dranyu@dranyu-desktop:~/file$ ls -l
总用量 12
drwxr-xr-x 2 dranyu dranyu 4096 2009-11-04 21:51 file2
-rw-r--r-- 1 dranyu dranyu    6 2009-11-04 21:50 h1
-rw-r--r-- 1 dranyu dranyu    7 2009-11-04 21:51 h2
dranyu@dranyu-desktop:~/file$ du
8    ./file2
20 .
```

可发现不带参数的 du 命令，显示了 file 目录与其子目录 file2 所占磁盘空间的大小。

例 13-17: 显示目录占用空间的总数

有时我们只需了解指定目录占用空间的总数情况，我们应使用-s 选项，代码如下所示：

```
dranyu@dranyu-desktop:~$ du -sh .
1.4G .
```

其中“.”表示当前目录，h 选项指明使用 MB 或 GB 单位显示磁盘空间，1.4GB 说明目前 dranyu 的主文件夹所占用的磁盘空间为 1.4GB。

13.5.4 挂载命令 mount

在 Ubuntu 中，文件系统需要被挂载之后才能浏览保存在其中的内容。Ubuntu 使用 mount 命令完成文件系统的挂载操作。

1. mount 的语法

mount 的语法结构为：

```
mount [-afFhnrvVw] [-L<标签>] [-o<选项>] [-t<文件系统类型>] [设备名] [加载点]
```

其中 mount 命令常用参数和选项如下所示：

- **-a:** 加载文件/etc/fstab 中设置的所有设备。
- **-f:** 不实际加载设备。可与-v 等参数同时使用以查看 mount 的执行过程。
- **-F:** 需与-a 参数同时使用。所有在/etc/fstab 中设置的设备会被同时加载，可加快执行速度。
- **-h:** 显示在线帮助信息。
- **-L<标签>:** 加载文件系统标签为<标签>的设备。
- **-n:** 不将加载信息记录在/etc/mtab 文件中。
- **-o<选项>:** 指定加载文件系统时的选项。

有些选项也可在/etc/fstab 中使用。这些选项包括：



- Async: 以非同步的方式执行文件系统的输入输出动作。
- Atime: 每次存取都更新 inode 的存取时间, 默认设置, 取消选项为 noatime。
- Auto: 必须在/etc/fstab 文件中指定此选项。执行-a 参数时, 会加载设置为 auto 的设备, 取消选取为 noauto。
- Defaults: 使用默认的选项。默认选项为 rw、suid、dev、exec、auto nouser 与 async。
- Dev: 可读文件系统上的字符或块设备, 取消选项为 nodev。
- Exec: 可执行二进制文件, 取消选项为 noexec。
- Noatime: 每次存取时不更新 inode 的存取时间。
- Noauto: 无法使用-a 参数来加载。
- Nodev: 不读文件系统上的字符或块设备。
- Noexec: 无法执行二进制文件。
- Nosuid: 关闭 set-user-identifier(设置用户 ID)与 set-group-identifier(设置组 ID)设置位。
- Nouser: 使一位用户无法执行加载操作, 默认设置。
- Remount: 重新加载设备。通常用于改变设备的设置状态。
- Ro: 以只读模式加载。
- Rw: 以可读写模式加载。
- Suid: 启动 set-user-identifier(设置用户 ID)与 set-group-identifier(设置组 ID)设置位, 取消选项为 nosuid。
- Sync: 以同步方式执行文件系统的输入输出动作。
- User: 可以让一般用户加载设备。
- -r: 以只读方式加载设备。
- -t<文件系统类型>: 指定设备的文件系统类型。

常用的选项说明有:

- Minix: Linux 最早使用的文件系统。
- ext2: Linux 目前的常用文件系统。
- msdos: MS-DOS 的 FAT。
- vfat: Windows85/98 的 VFAT。
- nfs: 网络文件系统。
- iso9660: CD-ROM 光盘的标准文件系统。
- ntfs: Windows NT 的文件系统。
- hpfs: OS/2 文件系统。Windows NT 3.51 之前版本的文件系统。
- auto: 自动检测文件系统。
- -v: 执行时显示详细的信息。
- -V: 显示版本信息。
- -w: 以可读写模式加载设备, 默认设置。

2. 操作实例

例 13-18: 列出所有已挂载的文件系统

`mount` 命令中在不使用任何参数的情况下查看当前系统中所有已挂载的文件系统，代码如下所示：

```
dranyu@dranyu-desktop:~$ mount
/dev/sda6 on / type ext3 (rw,relatime,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
...
/dev/sda1 on /boot type ext3 (rw,relatime)
securityfs on /sys/kernel/security type securityfs (rw)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,noexec,nosuid,nodev)
```

在操作结果中，每行记录表示一个文件系统的挂载情况。行中显示了文件系统的分区设备、文件系统的挂载点、类型等，如，“`/dev/sda6 on / type ext3 (rw,relatime,errors= remount-ro)`”记录，`/dev/sda6` 是分区设备名，`/`是挂载点，`ext3` 是文件系统类型。

例 13-19：挂载光盘

在挂载光盘之前一定要为光盘创建一个空白目录作为光盘的挂载点`/mnt/cdrom`，然后使用 `mount` 命令挂载光盘文件系统，代码如下所示：

```
dranyu@dranyu-desktop:~$ sudo su
[sudo] password for dranyu:
root@dranyu-desktop:/home/dranyu# mkdir /mnt/cdrom
root@dranyu-desktop:/home/dranyu# mount -t iso9660 /dev/cdrom /mnt/cdrom
```

例 13-20：挂载 FAT32 类型的文件系统

若在计算机上同时安装了 Windows 系统和 Linux 系统，要在 Linux 中可访问保存在 Windows 磁盘格式中的数据，则需要将 Windows 的分区挂载到 Linux 的目录下，具体操作的代码如下所示：

```
root@dranyu-desktop:/home/dranyu# mkdir /mnt/wind
root@dranyu-desktop:/home/dranyu# mount -t vfat /dev/sda2 /mnt/wind
```

其中`/dev/sda2` 文件系统的类型为 `vfat` 的磁盘分区设备，`/mnt/wind` 为挂载点目录。

13.5.5 卸载命令 `umount`

`umount` 命令是 `mount` 命令的反操作命令。光盘在没有卸载之前是不可弹出的。`umount` 命令的使用方法很简单，其语法为：

```
umount 挂载点
```

例 13-21：卸载光盘

卸载光盘可执行如下命令：

```
umount /mnt/cdrom
```

其中`/mnt/chrom` 参数为光盘挂载点目录。

例 13-22：卸载其他文件系统

```
umount /mnt/wind
```

注意，使用 `mount` 和 `umount` 的用户必须获得 root 管理权限。`umount` 命令不可卸载`/boot` 目录和有文件正在被读写的目录。



13.6 磁盘配额

Ubuntu 具有多用户多任务的特征，这就会出现多个用户同时登录到 Ubuntu 系统中使用其中资源的情况，若不合理安排管理，就会造成资源分配不均，不能充分利用资源的现象。为了避免某个用户在系统中占有过多的磁盘空间和资源的浪费，我们就需要使用磁盘配额来管理磁盘资源，限制用户使用磁盘容量的大小。

13.6.1 磁盘配额的简介

在 Ubuntu 中有两种方式实现磁盘配额，一种是基于文件节点（inode）的磁盘管理方式，还有一种是基于磁盘存储块（block）的磁盘管理方式。前者是通过限制用户能够拥有的最大文件数目来实现的，这里所说的文件不包括符号链接文件。后者是通过限制用户能够拥有的最大存储块数来实现的。

磁盘配额按可超越的程度分为软限制、硬限制。

- 软限制

用户所使用硬盘的容量超过软限制规定的大小时，用户仍然可以在不超出硬限制的范围内继续使用空间，同时用户会收到警告信息，但是如果在规定时间内用户仍未将使用空间恢复到正常状态，那么系统将不再允许该用户使用更多的空间。这里的规定时间我们称为宽限时间。

- 硬限制

硬限制是对空间使用的绝对限制，在任何情况下用户都不允许超过此限制。

13.6.2 配置磁盘配额

在默认情况下，Ubuntu 并没有安装磁盘配额工具。安装 quota 磁盘配额工具，可使用新立得软件包管理工具或在终端输入命令“sudo apt-get install quota”即可。安装 quota 工具后，我们就可对磁盘进行磁盘配额操作。

实现磁盘配额，我们可按照以下步骤进行：

- (1) 启动分区配额功能。
- (2) 生成配额文件。
- (3) 设置用户和组配额。
- (4) 启动磁盘配额服务。

为了说明磁盘配额的使用，首先使用 fdisk 分区工具创建一个新分区/dev/sda7，大小设置为 500MB，然后设置它的挂载点为/sda7，在系统中添加一个用户 sunny，将用户的主文件夹设置在/sda7 目录下，执行代码如下所示：

```
root@dranyu-desktop:/# mount /dev/sda7 /sda7  
root@dranyu-desktop:/# useradd -d /sda7/sunny -m sunny
```

接下来详细讲述以上步骤。

1. 启动配额功能

使用“`sudo su`”命令获取系统管理员身份，编辑系统配置文件`/etc/fstab`中相应分区的设置选项，若开启用户配额添加选项“`usrquota`”，用户组配额添加选项“`grpquota`”，中间以逗号隔开，代码如下所示：

```
/dev/sda7      /sda7      ext2      defaults,usrquota,grpquota 0 0
```

用户组配额用以限定一组用户共计可以使用的空间数，一般应用较少，用户在使用时可不设置`grpquota`选项。

完成设置之后，保存`/etc/fstab`文件，并重启计算机。那么以后系统启动后，`/dev/sda7`分区将自动挂载到`/sda7`，并且支持磁盘配额。若不希望重启，可简单点，即使用`mount`命令重新挂载`/dev/sda7`分区，使磁盘配额设置生效，命令如下所示：

```
root@dranyu-desktop:/# mount -o remount /sda7
```

为验证设置是否成功，可查看`/etc/mtab`文件，代码如下所示：

```
root@dranyu-desktop:/# cat /etc/mtab
/dev/sda6 / ext4 rw,relatime,errors=remount-ro 0 0
...
/dev/sda7 /sda7 ext2 rw,usrquota,grpquota 0 0
```

2. 生成配额文件

重新挂载`/dev/sda7`后，`/sda7`已能够使用磁盘配额，可是文件系统本身尚不能支持配额。这就需要使用`quotacheck`命令。`quotacheck`命令检测启动了配额的文件系统，并为每个文件系统建立当前磁盘用量表。该表会被用来更新操作系统的磁盘用量文件。同时，会在文件系统的根目录下创建或更新配额文件（`aquota.user`和`aquota.group`）。开始生成配额文件，执行命令如下所示：

```
root@dranyu-desktop:/sda7# quotacheck -cug /sda7
root@dranyu-desktop:/sda7# ls -la
总用量 32
drwxr-xr-x 4 root root 1024 2009-10-25 18:56 .
drwxr-xr-x 22 root root 4096 2009-10-25 17:48 ..
-rw----- 1 root root 7168 2009-10-25 18:56 aquota.group
-rw----- 1 root root 7168 2009-10-25 18:56 aquota.user
drwxr-xr-x 2 root root 12288 2009-10-25 15:13 lost+found
drwxr-xr-x 2 sunny sunny 1024 2009-10-25 17:50 sunny
```

其中选项`c`表示给文件系统创建配额文件；选项`u`表示给用户创建配额文件；选项`g`表示给用户组创建配额文件。上述代码中，使用了`ls`命令查看`/sda7`目录下的文件列表，发现多了`aquota.group`和`aquota.user`文件，说明配额文件已生成。

3. 设置配额

配额文件已有了，接下来，我们要为`sunny`用户分配磁盘空间。`aquota.group`和`aquota.user`文件的结构很复杂，我们不可直接编辑它们。因此，我们必须通过`edquota`命令来进行编辑。为`sunny`设置15块的磁盘空间，软限制为10MB，硬限制为15MB，而文件数设置就5个，软限制和硬限制不设置，执行如下代码：

```
root@dranyu-desktop:/sda7# edquota -u sunny
```

执行命令后显示`nano`编辑器。在`nano`编辑器中编辑数据，如图13-2所示。编辑完成后按下`Ctrl+O`组合键保存修改，然后按下回车键完成保存。

这里`blocks`表示存储块；`soft`表示软限制；`hard`表示硬限制；`inodes`表示文件节点。在



ext2 文件系统中每个存储块是 1KB 的大小。存储块段的 soft 和 hard 单位是字节。inodes 的单位为文件数。

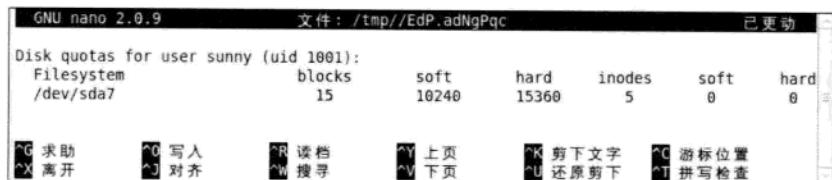


图 13-2 编辑 sunny 用户磁盘配额

对 sunny 用户分配配额后可使用 quota 命令查看配额信息，代码如下所示：

```
root@dranyu-desktop:/sda7# quota -u sunny
```

```
Disk quotas for user sunny (uid 1001):
```

文件系统块数量	配额	规限宽限期	文件结点	配额	规限宽限期
/dev/sda7	15	10240	15360	5	0

接着可以使用 edquota 命令创建宽限时间，使用选项 t 完成，输入命令如下所示：

```
root@dranyu-desktop:/home/dranyu# edquota -t
```

执行命令后显示 nano 编辑器，修改块存储的宽限时间为“14day”，文件节点的宽限时间为“8day”，如图 13-3 所示。

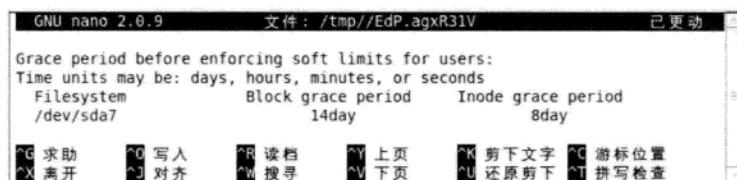


图 13-3 编辑宽限时间

补充说明

在使用 nano 编辑文件时，保存文件最好就按照默认文件保存。特别要注意的是，在设置宽限时间的时候，文件中不可出现“天”、“小时”等中文的单位，必须以“second”、“minute”、“hour” 和 “day” 作为计量单位。

4. 启动和关闭配额服务

使用命令 quotaon 来激活配额系统并允许配额检查；若关闭配额服务则使用命令“quotaoff”。启动和关闭配额服务的代码如下所示：

```
root@dranyu-desktop:/sda7/sunny/mdk# quotaon -avug
/dev/sda7 [/sda7]: group 配额已开启
/dev/sda7 [/sda7]: user 配额已开启
root@dranyu-desktop:/sda7/sunny/mdk# quotaoff -augv
/dev/sda7 [/sda7]: group 配额已关闭
/dev/sda7 [/sda7]: user 配额已关闭
```



13.7 小结

对于 Windows 上的用户或初接触操作系统的人来说，理解 Linux 的文件系统是一件不容易的事情。本章向用户介绍了文件系统的基础知识，介绍了 Ubuntu 中常用的文件系统类型，解释了 Ubuntu 中文件和目录的概念。为了能够让用户管理磁盘空间，本章还讲述了磁盘分区工具的使用方法和文件系统管理的基本命令，进而讲述了使用磁盘配额限制 Ubuntu 中多个用户的磁盘空间。

13.8 习题



1. 什么是文件系统？它包括哪些类型？Ubuntu 中支持哪些常用的文件系统？
2. Ubuntu 有哪几种文件类型？这些文件类型使用什么字母来表示？
3. 简述 Ubuntu 目录结构中各目录的含义。
4. 使用 fdisk 分区工具给磁盘划分一个大小为 500MB 的新分区，然后使用 parted 分区工具调整此分区的大小，将其调整为 1GB。
5. 使用 fdisk 分区工具创建一个大小为 500M 的新分区，然后修改其的文件系统类型。
6. 使用 df -h 命令查看文件系统的磁盘空间使用情况，并通过重定向符号将信息输出到 diskUsedInfo 文件。
7. 将光盘插入到光盘驱动器中，Ubuntu 会自动挂载光盘。此时，我们使用 umount 命令卸载光盘，然后使用 mount 命令重新挂载光盘。
8. 磁盘配额的作用是什么？使用磁盘配额有哪几个步骤？
9. 什么是软限制和硬限制？
10. 在当前系统中，创建一个名为 tom 的用户，然后使用磁盘配额为此用户分配 100 块的磁盘空间。
11. 修改用户 tom 的磁盘空间使用限制，为它分配 50 块的磁盘空间，并将硬限制修改为 1024 块。

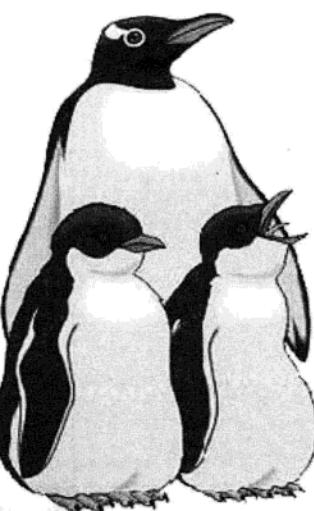
Linux

第14章

系统管理

在使用 Ubuntu 的过程中，对系统进行日常管理和维护是不可缺少的，而且是一项复杂的工作。对系统进行日常管理及维护，以保证系统安全、可靠、稳定地运行，确保用户合理有效地使用系统资源。

本章主要介绍用户与组管理，讲述进程管理，使用系统命令查看系统信息，最后讲述系统备份和恢复。



14.1 用户管理



Linux 是一个多用户的操作系统，它在管理多用户时就需要使用账号来识别登录到系统的每个用户。在 Ubuntu 中，用户可分为 3 种：超级用户、系统用户和普通用户。Ubuntu 的超级用户是 root 用户，他拥有最高权限，管理着系统的所有资源；系统用户是与系统服务相关的用户，其账号通常在系统安装及相关软件包安装时自动创建，一般不需要更改其设置；普通用户是登录和使用计算机来完成非管理任务的人，他的账号由超级用户或拥有管理用户账号权限的普通用户所创建。用户管理一般是指对普通用户的管理，其中包括了添加用户账号、设置用户登录口令、删除用户，以及修改用户属性等操作。



图 14-1 用户设置

Ubuntu 提供了用户管理的图形操作程序，使用系统管理员可方便地管理用户账号信息。接下来，我们使用用户管理图形操作程序创建用户账号、设置口令、删除用户账号，以及修改用户的属性。

选择 Ubuntu 顶部面板的“系统”|“系统管理”|“用户和组”菜单命令，弹出“用户设置”对话框，单击“解锁”按钮，在弹出的“认证”对话框中输入当前用户的密码，然后单击“认证”按钮。此时，“用户设置”对话框已解锁，如图 14-1 所示。

14.1.1 创立用户账户

使用用户管理图形操作界面创建用户账户方便、快捷，其具体操作如下所示：



图 14-2 设置用户账户的基本信息

(1) 打开“用户设置”对话框，然后单击“添加用户”按钮，弹出“新建用户账号”对话框。切换到“账户”标签页，在“用户名”文本框中输入用户名，如 sunny，在“真实姓名”文本框中输入“sunny”，从“配置文件”下拉列表中选择“Desktop User”选项设置创建用户时所使用的配置文件，在“密码”选项框中选中“手动设定密码”，然后在“用户密码”和“确认”文本框中输入 sunny 用户的密码，如图 14-2 所示。

(2) 为 sunny 添加管理用户账号的权限。在“新建用户账户”对话框中，选择“用户权限”选项卡，在罗列的管理操作列表中选择“管理用户”选项，设置此用户拥有管理用户的权限，如图 14-3 所示。

(3) 设置用户的主文件夹目录等高级设置。在“新建用户账户”对话框中，选择“高级”选项卡，在“主组”下拉列表中选择“dranyu”选项作为 sunny 所在的组，如图 14-4 所示。

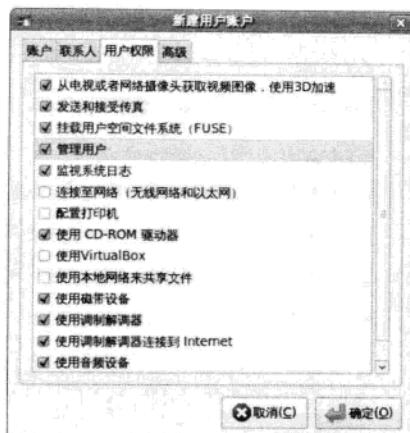


图 14-3 设置“管理用户”权限



图 14-4 设置“高级设置”

(4) 设置完毕之后，单击“确定”按钮完成新用户账号的添加操作，返回到“用户设置”对话框，如图 14-5 所示。



图 14-5 已添加了新用户账号 sunny

14.1.2 删除用户

在一个团队里，若有人由于某些原因离开了团队，他的 Ubuntu 账号不再使用，为了方便管理和节省资源，系统管理员就需要将他的账号删除。使用用户管理图形操作界面删除用户，操作简单，其方法为：在“用户设置”对话框中选中用户列表中的 sunny 记录，然后选择“删除”按钮，弹出提示框，如图 14-6 所示。单击“删除”按钮，删除 sunny 用户账号。

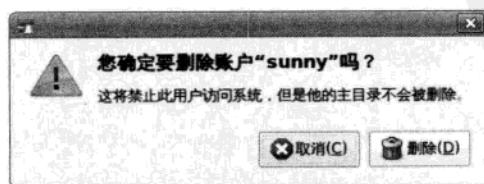


图 14-6 是否删除 sunny 账号

此时, sunny 的主文件夹仍然保存在/home 目录里, 我们可在终端上执行删除文件夹命令删除此文件夹, 命令如下所示:

```
sudo rm -r /home/sunny
```

14.1.3 修改用户属性

使用用户管理图形操作界面可修改用户的真实姓名、用户主文件夹、主组、所使用的 shell, 以及使用系统的权限, 还可修改用户的联系人信息。修改用户属性的具体操作步骤为: 在“用户设置”对话框中选中所需修改属性的 dranyu 用户账号, 然后单击“属性”按钮, 在弹出的“账户‘dranyu’的属性”对话框中, 设置相关的参数即可。此对话框中的设置参数含义与上一小节的“新建用户账户”对话框一致。

14.1.4 passwd 和 shadow 文件

在 Ubuntu Linux 系统中, 用户账号的基本信息存放在文件/etc/passwd 中。系统中的每个用户账号在此文件中都会有一行记录与之相对应, 查看/etc/passwd 文件中的内容可执行 cat 命令, 代码如下所示:

```
root@dranyu-desktop:/# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
syslog:x:101:102::/home/syslog:/bin/false
dranyu:x:1000:1000:dranyu,435,13078849161,13078849161:/home/dranyu:/bin/bash
sunny:x:1001:1000:dranyu,Sunny:/home/sunny:/bin/bash
...
uml-net:x:121:134::/home/uml-net:/bin/false
```

在此文件中, 每行记录的字段之间都使用“:”符号分隔, 总共包括了 7 个字段, 这些字段从左到右依次是: 登录账号、用户口令、用户的 UID 号、用户组的 GID 号、用户信息、用户的主文件夹, 以及登录 shell。这些字段的作用和含义如下所示:

- 登录账号

用户在登录系统时需要输入登录账号。在同一个系统中此登录账号是唯一的, 用户登录账号的使用有利于系统识别用户, 方便系统管理员辨别用户的身份。

- 用户口令

此项的值一般是“x”, 这是由于 Ubuntu 系统另外采用了/etc/shadow 文件来保存加密后的口令。这样做是为了加强系统的安全性, passwd 文件对所有用户来说都是可读写的, 对 root 管理员是可读写的; 而 shadow 文件只对 root 管理员是可读写的, 对其他的用户是不可读写的。

- 用户的 UID 号

UID 的值是一个数值, 系统中的每个用户都拥有唯一一个 UID 号。Ubuntu 中的系统的进程只能识别数字, 使用 UID 就是这个理由。可以说, 用户账号是在系统外标识用户, 而 UID 是在系统内标识用户。在 Ubuntu 中, 创建普通用户时, 系统会设置一个 UID 号, 此值是一个大于 1000 的数值。而 root 的 UID 值是 0, 在 1~100 之间的数值分配给 Ubuntu 中的系统用户, 在 100~1000 之间的数值是为其他的应用程序预留的, 如 mysql 数据库, 其 UID 为 114。

- 用户组的 GID 号



和 UID 相似, GID 是系统用户组的唯一标识。在 Linux 系统中一个用户可属于多个组, 一个组可包括多个用户。在此处 GID 只显示了用户属于的基本组。数值 0 表示是 root 的超级用户组的 GID, 在 1~100 范围内的数值为 Ubuntu 的系统 GID, 在 100~1000 范围内的数值是其他应用程序的 GID, 而普通用户的 GID 都不小于 1000。除了在 passwd 文件中指定其归属的基本组之外, 还在/etc/group 文件中指明一个组所包含的用户。

- 用户信息

这一字段包含有关用户的一些信息, 如用户的真实姓名、办公室地址、联系电话等, 信息之间使用 “,” 分隔。在 Linux 系统中, mail 和 finger 等程序利用这些信息来标识系统的用户。

- 用户的主文件夹

这一字段定义了个人用户的主文件夹, 当用户登录后, 他的 shell 将把该目录作为用户的工作目录。在 UNIX/Linux 系统中, 超级用户 root 的工作目录为 /root; 而其他个人用户在 /home 目录下均有自己独立的工作环境, 系统在该目录下为每个用户配置了自己的主目录。

- 登录 shell

这一字段指定用户的登录 shell。shell 是当用户登录系统时运行的程序名称, 通常是一个 shell 程序的全路径名, 如 /bin/bash。用户还可指定的常用 shell 包括: /bin/sh、/bin/csh 和 /bin/ksh 等。

在用户管理中, Ubuntu 除了使用 passwd 文件管理用户账户的信息外, 还使用了 shadow 文件。使用 cat 命令查看 shadow 文件中的内容结构, 代码如下所示:

```
root@dranyu-desktop:/# cat /etc/shadow
root:$6$Zsyy4/762N8/g/gk$/Nmy7DwkdXmhPIlQZ8AVdldzm/poTc/aBMvxJRFd7azAaEEHsZ0re333uJ
qIn05PvYiYLekKPC5Kzu.2dIFFH1:14547:0:99999:7:::
dranyu:$6$shLskZWPDKrEpk$WKylWqHgilZjH6aw4/VmKX9s4V24DaJ1Xx0yRjzcYMaL/yb7XoWvejmGI3
BnPTY0g3X8oyFicx8frTqbvuaZ0:14432:0:99999:7:::
sunny:!:14548:0:99999:7:::
```

在 /etc/shadow 文件中, 每一行就是用户账户的一条记录, 每行也同样使用 “:” 分隔。每行有 9 个字段, 分别表示: 登录账号、用户口令、上次更改口令的日期、口令还可被更改的天数、口令需要重新更改的天数、口令需要变更期限前的警告期限、账号失效期限、账号取消日期和保留字段。/etc/shadow 是很重要的文件, 不要删除此文件, 在编辑此文件的时候最好为此文件做个备份。

在用户管理中, 系统管理员可通过修改 /etc/passwd 和 /etc/shadow 这两个文件来添加普通用户。

补充说明

在设置 shadow 的日期时, 需要知道 Ubuntu 中日期的表示方法。在 Linux 系统中, Linux 会将 1970 年 1 月 1 日作为第 1 天, 那么 1971 年 1 月 1 日则为第 366 天。在查看 shadow 文件时, 有些记录的第 2 个字段也就是用户口令字段会出现 “*” 和 “!”。其中带有 “*” 符号的用户账号是不可从登录窗口登录的。而带有 “!” 符号的用户账号表示还没设置用户登录口令。

14.1.5 常用命令

在没有图形操作界面的情况下，系统管理员可使用系统提供的命令管理用户。

1. 添加用户账号命令

添加用户账号可使用 adduser 命令或 useradd 命令。此处说明 adduser 命令的使用方法，adduser 命令的基本语法如下：

```
adduser [-D] [选项] 用户名称
```

其中常用的选项如下：

- -d 目录：新账号每次登录时所使用的主文件夹。默认值为在/home 下创建一个与新用户同名的文件夹。如-d /home/sunny 表示为新建用户创建/home/sunny 主文件夹。
- -g 群组名：群组名须为现有存在的名称。群组数字也须为现有存在的群组。
- -G 群组列表：指定新建用户从属的群组列表。每个群组使用“,” 区隔开来，不可以夹杂空格符号。
- -s shell 的位置：新建用户登录后所使用的 shell 名称。预设为不填写，这样系统会帮你指定默认的登录 shell——/bin/bash。
- -u uid 使用者的 ID 值。必须为唯一的 ID 值，除非用-0 选项，数字不可为负值。预设为最小不得小于 999 而逐次增加。0~999 传统上是保留给系统账号使用。

举例如下：

- -M：表示创建新用户时不建立用户主目录。

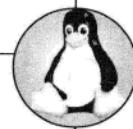
例 14-1：adduser 添加用户账号

在终端上使用 adduser 命令为系统添加一个名为 sun 的用户账号，代码如下所示：

```
root@dranyu-desktop:/# adduser sun      //注释：不使用其他选项时，  
adduser 命令进入交互操作模式  
正在添加用户 sun...  
正在添加新组 'sun' (1003)...  
正在添加新用户 'sun' (1002) 到组 'sun' ...  
创建主目录 '/home/sun'...  
正在从 /etc/skel 复制文件...  
输入新的 UNIX 口令： //输入新用户 sun 的密码，按下回车键  
重新输入新的 UNIX 口令： //再次输入新用户 sun 的密码，按下回车键  
passwd: 已成功更新密码  
正在改变 sun 的用户信息  
请输入新值，或直接敲回车键以使用默认值  
全名 []: sun //输入新建用户的全名，此处输入 sun  
房间号码 []: 435 //输入 sun 用户的房间号码，此处输入 435，按下回车键  
工作电话 []: 12345678912 //输入 sun 的工作电话，此处输入 12345678912，按下回车键  
家庭电话 []: 12345678912 //输入 sun 的家庭电话，此处输入 12345678912，按下回车键  
其他 []: 123456 //输入 sun 的袋子信息，此处输入 123456，按下回车键  
这个信息是否正确？ [Y/n] y //信息确认无误，输入 y，按下回车键。
```

在“输入新的 UNIX 口令：”提示输入 sun 的口令，在“重新输入新的 UNIX 口令：”提示重新输入 sun 的口令。其他信息可自行设置，完成设置之后，在“这个信息是否正确？[Y/n]”提示符后输入 y 完成 sun 用户的添加。

而使用 useradd 命令就没有使用 adduser 命令那么方便了，使用 useradd 命令创建的用户，并没有设定用户口令，还要借助 passwd 命令来设置用户口令。



2. 设置口令的命令

在 Ubuntu 中用 passwd 命令管理用户口令，它的基本语法如下：

```
passwd [选项] 用户名
```

其常用的选项如下：

- -l：此项表示锁定已经命名的账户名称，只有具备超级用户权限的使用者方可使用。
- -u：此项表示解开账户锁定状态，只有具备超级用户权限的使用者方可使用。
- -x：此项设置密码使用的最长时间，单位为天，只有具备超级用户权限的使用者方可使用此选项。
- -n, --minimum=DAYs：最小密码使用时间（天），只有具备超级用户权限的使用者方可使用。
- -d：删除使用者的密码，只有具备超级用户权限的使用者方可使用。
- -S：检查指定使用者的密码认证种类，只有具备超级用户权限的使用者方可使用。

例 14-2：为新建用户分配口令

先使用 useradd 创建 sunny 用户，再使用 passwd 命令设置用户 sunny 的口令。

```
root@dranyu-desktop:~# useradd sunny  
root@dranyu-desktop:~# passwd sunny //注释：不使用其他选项设置 sunny 用户的密码  
输入新的 UNIX 口令： //输入 sunny 用户的新密码，按下回车键  
重新输入新的 UNIX 口令： //再次输入 sunny 用户的新密码，按下回车键  
passwd: 已成功更新密码
```

在 Ubuntu 中，默认情况下 root 管理员账号是不可登录的。这是考虑到 root 身份的特殊性问题，由于 root 在系统中拥有最高的权限，为了防止使用 root 身份而造成不可弥补的误操作，Ubuntu 就将 root 账号锁定了。若一定要使用则可使用 sudo su 获得 root 的管理员权限，或使用 passwd 命令为 root 设置用户口令，如例 14-3 所示。

例 14-3：开启 root 登录功能

```
dranyu@dranyu-desktop:~$ sudo passwd root  
输入新的 UNIX 口令： //输入 root 的新密码，按下回车键  
重新输入新的 UNIX 口令： //再次输入 root 的新密码，按下回车键  
passwd: 已成功更新密码
```

若要锁定 sunny 用户，限制 sunny 命令的登录可使用-l 选项执行 passwd 命令，如例 14-4 所示。

例 14-4：锁定用户

执行过程如下所示：

```
root@dranyu-desktop:~# passwd -l sunny //注释：使用-l 选项锁定用户 sunny  
密码未更改。  
root@dranyu-desktop:~# cat /etc/shadow | grep sunny //显示密码影子文件中 sunny 用户的记录  
sunny:!$6$0yf1PS3C$4ZtwrMsbPHpWRur3FXVZfnJfcD/zsiQSz0jXgtdQcq4xmQjpSF5hfHDC1psX9I80  
THxCoCovBO.hd4AShyz9f0:14548:0:99999:7:::
```

结果显示 sunny 用户在/etc/shadow 文件中，他的用户口令字段值的第一字符为“!”表示目前 sunny 不可登录，如例 14-5 所示。

例 14-5：为已锁定用户解锁

为 sunny 用户解锁，操作过程如下所示：

```
root@dranyu-desktop:~# passwd -u sunny //使用-u 选项对锁定的 sunny 用户进行解锁  
密码未更改。
```

```
root@dranyu-desktop:~# cat /etc/shadow | grep sunny //显示 shadow 文件中 sunny 密码记录
sunny:$6$0yflPS3C$4ZtWrMsbPHpWRur3FXVZfnJfcD/zsiQSz0jXgtdQcq4xmQjpSF5hfHDClpsX9I8OT
HxCcOvBO.hd4AShyz9f0:14548:0:99999:7:::
```

与例 14-4 相比，在 sunny 用户账号记录行中用户口令字段前面的“!”符号没有了。

3. 删除用户账号命令

若用户不再使用系统时，我们可使用 userdel 命令将此用户账号删除。userdel 命令的语法格式为：

```
#userdel [选项] 用户名
```

在不使用选项的时候 userdel 命令只删除用户账号、用户组信息和个人信息，而保留了用户的主文件夹，如例 14-6 所示。若在删除用户账号时同时删除用户的主文件夹，在使用 userdel 命令的时候添加-r 选项，如例 14-7 所示。

例 14-6：删除用户时不删除主文件夹

```
root@dranyu-desktop:~# userdel sunny
root@dranyu-desktop:~# ls -l /home //显示/home 目录下用户的主文件夹
总用量 8
drwxr-xr-x 70 dranyu dranyu 4096 2009-10-30 23:44 dranyu
drwxr-xr-x 2 1001 dranyu 4096 2009-10-31 10:26 sunny //sunny 用户的文件夹还存在
```

例 14-7：删除用户账号时同时删除主文件夹

```
root@dranyu-desktop:~# userdel -r sunny //使用-r 选项删除用户账号
root@dranyu-desktop:~# ls -l /home
总用量 8
drwxr-xr-x 70 dranyu dranyu 4096 2009-10-30 23:44 dranyu
```

4. 修改用户属性命令

在使用添加用户账号命令的时候，系统管理员可指定新账号的相关用户信息，在添加用户账号之后，系统管理员可使用 usermod 命令修改用户账号的信息，usermod 命令的基本语法如下：

```
usermod [选项] 用户名
```

usermod 中常用的参数如下：

- -d 用户文件夹：更新使用者新的登录目录。
- -g 群组名：更新使用者新的起始登录群组。群组名须已存在。群组 ID 必须参照既有的群组。群组 ID 预设值为 1。
- -G 群组名 1,...：定义使用者为一群 groups 的成员。每个群组使用“,”区隔开来，不可以夹杂空白符号。
- -s shell 指定新登录 shell。如此栏留白，系统将选用系统预设 shell。
- -u 用户 ID：修改用户的 ID 值。必须为唯一的 ID 值。

例 14-8：修改 sunny 用户 shell 属性

sunny 用户是使用 useradd 命令创建的，其默认使用的 shell 为/bin/sh。现在我们使用 usermod 命令修改 sunny 用户的登录 shell，代码如下所示：

```
root@dranyu-desktop:~# cat /etc/passwd | grep sunny //查看密码文件中 sunny 的记录信息
sunny:x:1003:1004::/home/sunny:/bin/sh
root@dranyu-desktop:~# usermod -s /bin/bash sunny
root@dranyu-desktop:~# cat /etc/passwd | grep sunny //查看密码文件中 sunny 的记录信息
sunny:x:1003:1004::/home/sunny:/bin/bash
```

例 14-9：修改 sunny 用户的 UID、用户名和 GID 属性



将 bob 的 UID 修改为 1005，且将其用户名改为 sun，再把它的主组改为 root 组，组值为 0，代码如下所示：

```
root@dranyu-desktop:~# usermod -u 1005 -l sun -g 0 sunny  
root@dranyu-desktop:~# cat /etc/passwd | grep sun  
sun:x:1005:0::/home/sunny:/bin/bash
```

14.2 组管理



每个用户都属于特定的用户组，而用户组是一些具有相同特性的用户集合。每个用户在 /etc/passwd 文件中指定了一个他所从属的基本组，而在 /etc/group 文件中指定了组的成员，也就是说在 Ubuntu 中用户有一个基本组，同时他还可能属于其他的组。

选择 Ubuntu 顶部面板“系统”|“系统管理”|“用户和组”菜单命令，弹出“用户设置”对话框，单击“管理组”按钮，弹出“组设置”对话框，如图 14-7 所示。

在此对话框中，系统管理员可添加用户组、修改组的属性，以及删除组。

14.2.1 用户组的添加和删除

使用图形操作界面添加组的具体操作步骤如下所示：

- (1) 在“组设置”对话框中单击“添加组”按钮，弹出“新建组”对话框，如图 14-8 所示。
- (2) 在“组名(名)”文本框中输入新建群组名，如 project1，在“组的 ID(I)”文本框中选择 1005 作为 GID，在组成员列表中选择“dranyu”、“sunny”和“root”选项作为 project1 组的成员。
- (3) 设置完毕后，单击“确定”按钮，添加 project1 组，可在“组设置”对话框上的组列表中查看到 project1 记录，表示新组 project1 已添加成功，如图 14-9 所示。



图 14-7 “组设置”对话框

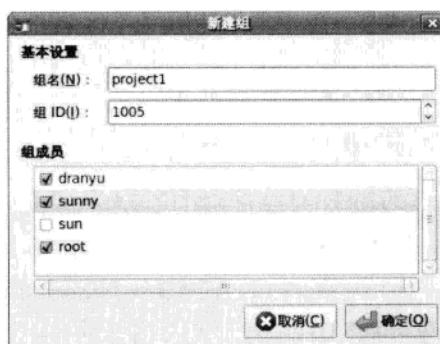


图 14-8 设置新建组



图 14-9 添加新组 project1

系统管理员还可以在模拟终端上使用 groupadd 命令添加一个组。使用 groupadd 命令添加用户组，其基本语法如下：

```
groupadd [-g gid[-o]] [-r] [-f] 用户组名
```

其中选项的含义如下。

- -g：此选项后接一个 GID 的值，用来指定新用户组的组标识。
- -o：此选项是和-g 选项一起使用的。如果使用该选项，则允许使用相同的组标识。
- -r：此选项指明要生成一个系统组。
- -f：此选项是一个强制选项。

例 14-10：使用 groupadd 添加用户组

要添加一个用户组 project2，并指定它的 GID 为 1007，代码如下所示：

```
root@dranyu-desktop:~# groupadd -g 1007 project2
root@dranyu-desktop:~# cat /etc/group | grep project
project1:x:1005:dranyu,root,sunny
project2:x:1007:
```

当一个软件开发项目结束之后，此时可能需要删除为此项目而创建的组。可使用图形操作界面，也可使用 groupdel 命令删除用户组，其操作很简单。

使用图形操作界面的方法为：在“组设置”对话框的组列表框中选中需要删除的组，然后单击“删除”按钮即可删除指定的组。

groupdel 命令格式为：

```
groupdel 组名
```

例 14-11：使用 groupdel 命令删除组

删除 project2 用户组的操作如下所示：

```
root@dranyu-desktop:~# groupdel project2
root@dranyu-desktop:~# cat /etc/group | grep project
project1:x:1005:dranyu,root,sunny
```

14.2.2 修改组的属性

当建立好一个组之后，需要对组进行修改，可使用图形操作界面，也可使用 groupmod 命令修改组的属性。

使用图形操作界面修改组属性的方法为：在“组设置”对话框的组列表中选中所需修改的组，然后单击“属性”按钮，弹出组属性编辑对话框，修改相关的参数之后，单击“确定”按钮完成设置。

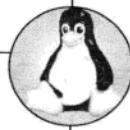
使用 groupmod 命令修改组的属性的语法如下所示：

```
groupmod [-g GID[-o]] [-n group_name] 组名
```

除了-n 选项，其他选项的含义与 groupadd 命令中相应的选项含义一样。-n 选项指定将原来的组名修改为指定的组名。

例 14-12：将组 project1 改名为 project2，组的 GID 改为 1008，代码如下所示：

```
root@dranyu-desktop:~# groupmod -g 1008 -n project2 project1
root@dranyu-desktop:~# cat /etc/group | grep project
project2:x:1008:dranyu,root,sunny
```



14.2.3 /etc/group 文件

Ubuntu 中的组信息保存在/etc/group 文件中。每一行表示一条组的记录，每行中有 4 个字段并以“:”分隔，这 4 个字段分别是组名、组的口令、组的标识 GID 和组的成员。在组成员字段，每个成员间用逗号隔开。

例 14-13：查看/etc/group 文件的内容

使用 cat 命令查看 group 文件的内容，代码如下所示：

```
root@dranyu-desktop:~# cat /etc/group
root:x:0:
...
admin:x:121:dranyu
dranyu:x:1000:
...
vboxusers:x:125:dranyu
...
project2:x:1008:dranyu,root,sunny
```

系统管理员在管理组的时候，可根据需要编辑此文件。添加组，则可在此文件的后面插入一行格式相同的组记录；删除组，可将所要删除的组记录删除；修改组，可直接编辑所要修改的组记录。

14.3 进程管理



进程管理是操作系统中重要的一部分。在一个多进程的操作系统中，一块时间片内可以有多个进程同时执行。这样可避免较快速的 CPU 等待较慢的 I/O 设备的情况，提高 CPU 的利用率，从而提高操作系统的性能。了解进程有助于用户管理系统中同时执行的进程。

14.3.1 进程的概念

进程是指正在运行的程序，是动态的，例如每个用户执行的任务、每个系统服务，都可以称为进程。进程虽由程序产生，却有别于程序。程序是静态的，由多条指令组成，不占用系统的运行资源；而进程是随时都可能发生变化的、动态的并且需要使用系统的运行资源，一个程序可产生多个进程。

进程有 3 种基本状态，包括：运行状态、就绪状态和阻塞状态。进程的动态性就是体现在这 3 种基本状态之间的相互转换。

- 运行状态

在此状态下，当前进程已分配到 CPU 的处理时间片，此时进程在计算机中运行。

- 就绪状态

此状态表示进程已具备运行的条件，可并没分配到 CPU 的处理时间片。

- 阻塞状态

此状态表示进程因等待某种事件的发生而暂时不具备运行的条件。

这三种状态之间可进行相互转换，其包括：由就绪状态转换成运行状态；由运行状态转

换成阻塞状态；由阻塞状态转换成就绪状态；由运行状态转换成就绪状态。

进程实体通常由程序、数据集合和进程控制块 3 部分组成。而其中进程控制块是用来描述进程当前的状态、本身特性的数据结构，有时也称为进程描述块。

在 Linux 系统中，进程可分为系统进程和用户进程两大类。系统进程与终端无关，不由用户启动，通常也称为守护进程，是核心专用的特殊程序。用户进程是指由用户启动的进程，同时也可由用户关闭。

14.3.2 进程显示和删除

在 Ubuntu 中，提供了多种命令查看和删除进程，如 ps、top 和 kill。Ubuntu 也提供了进程的图形管理程序，将在后续章节 14.3.4 中讲述。此节讲述 ps、top 和 kill 命令的简单使用。

1. ps 命令

ps 命令显示进程的功能十分强大，它主要显示进程的运行状态信息，显示进程占用资源的信息等。其显示的信息是在特定的某个时间点上进程的状态。ps 命令的语法如下所示：

ps [选项]

常用的选项如下所示。

- -l：显示进程的详细信息。
- -a：显示所有用户的所有进程的信息。
- u：显示当前用户的进程的状态信息。
- -x：显示与控制终端无关的进程。
- -f：以目录树的方式显示进程。

例 14-14：不带参数的 ps 命令

直接输入 ps 命令，执行 ps 命令结果如下所示：

```
dranyu@dranyu-desktop:~$ ps
  PID TTY      TIME CMD
 5329 pts/0    00:00:00 bash
 5346 pts/0    00:00:00 ps
```

其中 PID 字段表示进程标识，TTY 字段表示登录的终端机号，TIME 字段表示此进程所消耗的 CPU 时间，CMD 字段表示正在执行的命令及进程名称。

例 14-15：使用-u 选项显示进程

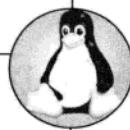
使用-u 选项的主要功能是将进程数据以用户的格式来显示，执行命令和结果显示如下所示：

```
dranyu@dranyu-desktop:~$ ps u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
dranyu     5329  0.0  0.3   6652  3724 pts/0      Rs Nov05   0:00 bash
dranyu     7874  0.0  0.0   2768  1036 pts/0      R+  00:37   0:00 ps u
```

有些选项的含义与例 14-14 相同，其他的说明为：USER 字段显示进程的所有者；%CPU 字段表示 CPU 的使用率百分比；%MEM 字段表示内存使用率的百分比；VSZ 字段表示进程所占用的虚拟内存的大小；RSS 字段表示进程占用的物理内存大小；STAT 表示进程的状态；START 字段表示进程被启动的开始时间；COMMAND 字段表示显示的启动此进程的进程名称。

STAT 的值有 4 种，表示进程的状态如下所示。

- R：表示此进程正在运行。



- s: 表示此进程正在休眠，可以被其他进程唤醒。
- T: 表示此进程已经停止或者是正在进行追踪。
- Z: 表示此进程是僵尸进程，即此进程已经停止，但是其父进程无法正常关闭它。

2. top 命令

ps 命令显示的是某一时间进程的运行状态，而 top 命令显示的是静态的信息；Ubuntu 中提供了动态显示进程信息的命令，就是 top 命令。使用 top 命令能够每隔一段时间对进程重新显示，主要显示 CPU、内存的使用率，以及相关进程的信息等。它的基本语法如下所示：

```
top [选项]
```

常用的选项如下所示。

- -d: 指定每两次屏幕信息刷新之间的时间间隔。
- -q: 此选项将使 top 无任何延迟地进行刷新。
- -S: 制定累计模式。
- -s: 设置刷新率。
- -I: 此选项使 top 不显示任何限制或者僵尸进程。
- -c: 显示整个命令行而不只是显示命令名。
- -u: 设置显示用户的进程。

例 14-16：不带参数的 top 命令

Ubuntu 下不带参数的 top 命令会按照默认设置执行，其中默认每隔 3 秒钟进行一次刷新，执行不带参数的 top 命令，执行如下所示：

```
top
```

执行命令后显示 top 的字符操作界面如图 14-10 所示。

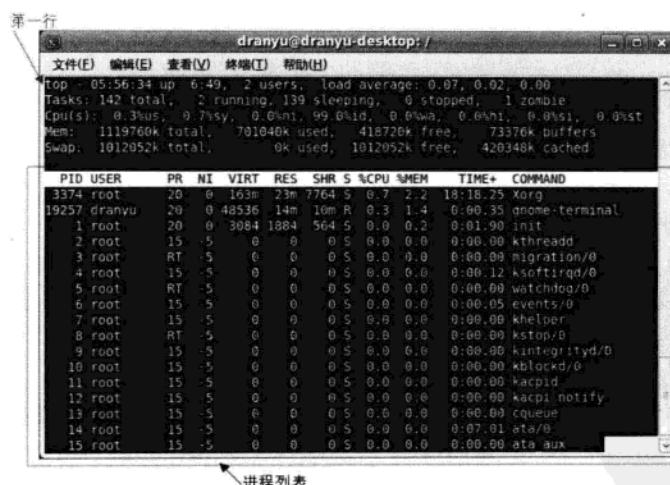


图 14-10 top 字符操作界面

top 命令显示的项目比较多，我们来说明其中的部分内容。

- 第一行的项目中包括了系统运行的总时间、当前系统的用户人数和每户平均负载。

它们之间是使用“,”分隔开。

- 第二行的项目显示了进程统计情况，依次显示了进程的总数、运行进程数、休眠进程数、终止进程数和僵尸进程数。
- 第三行的项目显示了系统 CPU 的使用状况，其中“0.3%us”表示用户的进程占用 CPU 的比率，“0.7%sy”表示系统进程占用 CPU 的比率。
- 第四行的项目显示了内存的使用情况，依次显示了内存总数、已用的内存数、空闲的内存数和缓存使用的内存数。
- 第五行的项目显示了交换分区状态，各字段的含义依次为交换分区的总数、当前已使用的空间数、当前空闲的空间数和高速缓存容量。

进程列表中显示了当前进程的详细信息。

补充说明

在 top 的字符操作界面中，可使用其他的 top 内置命令控制进程信息显示的方式。如 t 命令切换第二行显示进程统计情况，l 命令切换第一行显示系统总体情况，P 命令使进程列表按 CPU 的使用率的大小进行排序，m 命令切换显示内存信息等。

3. kill 命令

当需要结束一个正在前台运行的进程时，我们可以使用“Ctrl+C”组合键来完成。若要结束一个在后台运行的进程时，我们就需要使用 kill 命令来结束进程。kill 命令杀死进程是 kill 命令向 Linux 系统内核发送一个系统操作信号和某个程序的进程标志号，然后系统内核就对进程标志号指定进程进行操作。dranyu@dranyu-desktop:\$ kill -l

kill 命令的基本语法如下所示：

```
kill [-信号名 | -s 信号标识] 进程标识号
```

kill 命令的常用选项如下所示。

- s：指定需要送出的信号。既可以是信号名也可以是对应数字。
- p：指定 kill 命令只是显示进程的 pid，并不真正送出结束信号。
- l：显示信号名称列表。信号名称列表也可以在/usr/include/linux/signal.h 文件中找到。

例 14-17：显示系统的系统操作信号信息

```
dranyu@dranyu-desktop:$ kill -l //使用-l 选项，显示系统操作信号列表
1) SIGHUP    2) SIGINT    3) SIGQUIT    4) SIGILL
5) SIGTRAP   6) SIGABRT   7) SIGBUS     8) SIGFPE
9) SIGKILL   10) SIGUSR1   11) SIGSEGV   12) SIGUSR2
13) SIGPIPE   14) SIGALRM   15) SIGTERM   16) SIGSTKFLT
17) SIGCHLD   18) SIGCONT   19) SIGSTOP   20) SIGTSTP
21) SIGTTIN   22) SIGTTOU   23) SIGURG    24) SIGXCPU
25) SIGXFSZ   26) SIGVTALRM 27) SIGPROF   28) SIGWINCH
29) SIGIO     30) SIGPWR    31) SIGSYS   34) SIGRTMIN
35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3 38) SIGRTMIN+4
39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12
47) SIGRTMIN+13 48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14
51) SIGRTMAX-13 52) SIGRTMAX-12 53) SIGRTMAX-11 54) SIGRTMAX-10
55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7 58) SIGRTMAX-6
59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
```



63) SIGRTMAX-1 64) SIGRTMAX

其中数字为系统操作信号的标识,而数字后的字符串表示操作信号的名称。如9) SIGKILL 表示系统操作信号的标识为9,其的名称为SIGKILL。

例 14-18: 杀死后台进程

使用&符号将 Gedit 应用程序放到后台运行,然后执行 ps 命令查看 Gedit 程序的进程标识号,最后使用 kill 杀死 Gedit 进程,操作过程如下所示:

```
dranyu@dranyu-desktop:/$ Gedit & //启动 Gedit 应用程序
[1] 24186 //Gedit 应用程序的进程 ID
dranyu@dranyu-desktop:/$ ps //显示当前用户运行的应用程序信息
  PID TTY      TIME CMD
19261 pts/0    00:00:00 bash
24186 pts/0    00:00:00 Gedit //Gedit 应用程序的进程 ID
24196 pts/0    00:00:00 ps
dranyu@dranyu-desktop:/$ sudo kill 24186
[sudo] password for dranyu:
[1]+  已终止                  Gedit
```

在“sudo kill 24186”命令中,并没有指定要发送信号的,kill 会默认发送杀死信号。若这样,Kill 命令还不能将应用程序杀死,用户可使用 kill 命令向系统发送信号,执行命令如下所示:

```
dranyu@dranyu-desktop:/$ sudo kill -s 9 25332
[sudo] password for dranyu:
[1]+  死亡                  Gedit
```

使用发送信号的方式杀死进程表示强行关闭进程。

14.3.3 进程调度

在第 10 章中,我们讲述了进程在前后台的切换方法。在这一小节中,我们将讲述有关系统的任务自动调度的内容。这种任务自动调度是指在指定的时间执行指定的一些操作,或者周期性地执行指定的任务。需要自动调度的任务通常可分为两类:系统任务和单个用户任务。这些任务都可由 crontab 命令来完成。

Cron 是 Ubuntu 中一个重要的守护进程,它不仅属于驻留服务的一种,而且提供定时器的功能。因此,我们可通过它来执行周期性的任务。命令 crontab 可以为我们建立 cron 任务,并帮助我们编辑和删除 cron 任务,而且每个用户都有一个属于自己的 crontab 文件用来保存发挥自己的 cron 任务。

crontab 命令的基本语法如下:

```
crontab [ -u user ] file
crontab [ -u user ] { -l | -r | -e }
```

crontab 命令的常用选项如下所示。

- **-t:** 执行文字编辑器来设定时程表,内定的文字编辑器是 VI,如果你想用别的文字编辑器,则请先设定 VISUAL 环境变数来指定使用哪个文字编辑器(比如说 setenv VISUAL joe)。
- **-r:** 删除目前的时程表。
- **-l:** 列出目前的时程表。
- **-u user:** 对指定用户的 cron 任务进行操作。

- -e: 建立或者编辑 cron 任务。

例 14-19: 查看 crontab 任务进程表的内容格式

使用 cat /etc/crontab 命令, 查看 cron 任务的编辑格式, 代码如下所示:

```
root@dranyu-desktop:/# cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab` command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user    command
17 * * * *    root    cd / && run-parts --report /etc/cron.hourly
25 6 * * *    roottest -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.daily )
47 6 * * 7    roottest -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.weekly )
52 61 * * *   roottest -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.monthly )
#
```

时程表的格式如下:

```
m h dom mon dow user command
```

其中 m 表示分钟, h 表示小时, dom 表示月份中的第几日, mon 表示月份, dow 表示星期中的第几天, command 表示要执行的程序。

当 m 为 * 时表示每分钟都要执行 command, f2 为 * 时表示每小时都要执行程序, 其余类推。

当 m 为 a-b 时表示从第 a 分钟到第 b 分钟这段时间内要执行, h 为 a-b 时表示从第 a 到第 b 小时都要执行, 其余类推。

当 m 为 */n 时表示每 n 分钟个时间间隔执行一次, h 为 */n 表示每 n 个小时时间间隔执行一次, 其余类推。

当 m 为 a, b, c,... 时表示第 a, b, c,... 分钟要执行, h 为 a, b, c,... 时表示第 a, b, c,...个小时要执行, 其余类推。

1. 建立 cron 任务

说明了时程表的格式后, 我们来看看如何建立一个 cron 任务。建立 cron 任务的具体操作步骤如下所示:

(1) 在终端输入如下命令:

```
root@dranyu-desktop:~# crontab -e //使用-e 选项, 创建用户 root 的任务
no crontab for root - using an empty one //第一次执行操作并不拥有 root 用户的 crontab 配置文件
Select an editor. To change later, run 'select-editor'.
1. /usr/bin/vim.tiny
2. /bin/ed
3. /bin/nano      <---- easiest
4. /usr/bin/vim.nox
5. /usr/bin/vim.basic
```



Choose 1-5 [3]: //在此输入 1-5 数字选择编辑器, 不输入表示使用默认设置, 此处默认设置为 3, 即选择 nano 编辑器

(2) 在弹出的 nano 编辑界面中按照时程表的格式输入 cron 任务, 如: “5 * * * * root ls -al > ~/test.txt”, 如图 14-11 所示。



图 14-11 添加 cron 任务

(3) 编辑完成后, 按下“Ctrl+O”组合键保存任务, 然后按下“Ctrl+X”组合键退出编辑器, 在终端显示如下代码:

```
crontab: installing new crontab
```

这表示 cron 任务已添加。

在编辑 cron 任务时, 可使用一些特殊的字符如下。

- 逗号 “,”: 表示不连续的时间点, 如 “1,15”若在第一列表示第 1 分钟和第 15 分钟。
- 横线 “-”: 表示一个时间段。
- 斜杠 “/”: 使用 “/数字” 表示每隔多长时间, 时间由数字指定。

下面举例说明 cron 任务的写法。

例 14-20: 每小时的第 15 分钟和第 30 分钟执行 ls -al >~/test.txt, 其 cron 任务的写法如下所示:

```
15,30 * * * * ls -al > ~/test.txt
```

例 14-21: 每小时的第 1~5 分钟中每分钟执行一次/usr/bin/back.sh 文件, 其 cron 任务的写法如下所示:

```
1-5 * * * * /usr/bin/back.sh
```

例 14-22: 每隔 15 分钟执行一次 rm -rf /tmp 命令, 其 cron 任务的写法如下所示:

```
*/15 * * * * rm -rf /tmp
```

例 14-23: 每天 12:30 执行/usr/local/etc/rc.d/lighttpd restart 命令重启 appach 服务器, 其 cron 任务的写法如下所示:

```
30 12 * * * /usr/local/etc/rc.d/lighttpd restart
```

例 14-24: 每星期一的 9:50 执行/usr/bin/data.sh 文件, 其 cron 任务的写法如下所示:

```
50 9 * * 1 /usr/bin/data.sh
```

例 14-25: 每年的 8 月 15 日 15:25 执行 ls 命令, 其 cron 任务的写法如下所示:

```
25 15 15 8 * ls
```

2. 删除 cron 任务

当一个 cron 任务结束后, 就需要将此 cron 任务删除, 让它不再运行。使用 crontab -r 命令可删除 cron 任务, 如例 14-26 所示。

例 14-26：删除当前用户的 cron 任务

执行删除命令，然后使用-l 选项列出当前用户的 cron 任务，显示结果如下所示：

```
root@dranyu-desktop:~# crontab -r //删除 cron 任务
root@dranyu-desktop:~# crontab -l //罗列 cron 任务
no crontab for root
```

最后一行表示 root 用户没有设置 cron 任务。

3. 启动 cron 任务

若用户以 crontab -e 命令创建的 cron 任务，那么在设置完毕之后，系统会将用户的设置内容写入 /var/spool/cron/crontabs/ 目录下并以每个用户名命名设置文件，同时自动检查并启动调度任务。启动后，系统会自动在配置文件 /var/spool/cron/crontabs/root 中添加启动信息文本，此时 root 文件的内容如下所示：

```
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.PJDZTr/crontab installed on Fri Nov 6 16:35:35 2009)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
5 * * * * ls -al >~/test.txt
```

若系统管理员是使用 vi 等其他编辑器编辑配置文件，那么需要使用 crontab 命令来启动任务调度。如使用 Gedit 编辑 /var/spool/cron/crontabs/root 配置文件后，启动任务调度，需要执行的命令如下所示：

```
root@dranyu-desktop:~# crontab /var/spool/cron/crontabs/root
```

14.3.4 系统监视器管理进程

使用图形操作界面可方便管理进程。选择 Ubuntu 顶部面板的“系统”|“系统管理”|“系统监视器”菜单命令，弹出“系统监视器”对话框，选择“进程”选项卡，显示界面如图 14-12 所示。在进程列表中，表头上部分字段含义如下所示。

- %CPU：表示进程所占用 CPU 资源的百分比。
- Nice：表示进程的优先级。
- ID：表示进程的标识。
- 命令行：表示进程的静态程序的位置和程序的参数。
- 内存：表示进程所使用的内存资源的大小空间。
- 等候频道：表示进程目前所在的等待频道。

在列表中，可查看系统中的进程数据，了解系统的平均负载等。在此界面中，用户可“杀死”进程、停止进程、结束进程和调整进程的优先值。用鼠标右键单击空白区域，弹出快捷菜单如图 14-13 所示。在弹出的快捷菜单中选择相应的菜单就可使用 kill 命令完成操作。

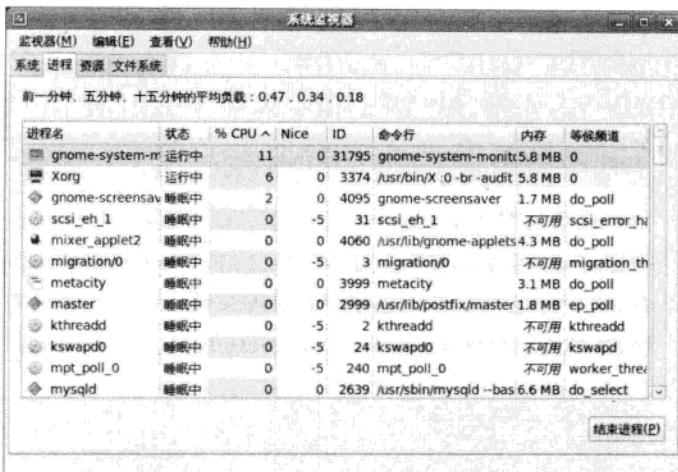


图 14-12 系统监视器

图 14-13 弹出的快捷菜单

14.4 查看系统信息

在系统管理中，了解系统信息有利于管理系统的资源，使系统管理员能够方便维护系统。Ubuntu 提供了了解系统信息的命令和图形程序，帮助系统管理员完成系统管理任务。通过查看系统的硬件信息，系统管理员可了解到目前系统中硬件的连接情况；通过阅读系统日志，系统管理员可检查系统错误产生的原因和系统受到攻击的踪迹；了解系统内核的模块加载信息，系统管理员可自行优化系统的运行。

14.4.1 系统硬件信息

在 Ubuntu 中，系统管理员可使用 hardinfo 程序查看系统硬件信息。hardinfo 用来采集系统硬件和操作系统方面的相关信息、执行标准检查程序，并能够生成可供打印的 HTML 或者纯文本格式报告。目前，此程序能够识别 PCI、ISA PnP、USB、IDE、SCSI、串口和并口设备。

由于 Ubuntu 默认不安装 hardinfo，所以在使用 hardinfo 工具之前，我们需要安装这一工具。打开新立得软件包管理器，找到 hardinfo 软件包，然后安装此软件包，或在终端输入如下命令：

```
sudo apt-get install hardinfo
```

安装 hardinfo 之后，选择顶部面板上的“系统”|“首选项”|“System Profiler and Benchmark”菜单命令，启动 hardinfo 应用程序，单击左边的“Summary”选项，显示系统信息摘要，如图 14-14 所示。

使用 hardinfo 可生成系统信息报告，具体操作步骤如下：

(1) 单击快捷工具栏上的“Generate Report”按钮，弹出“Generate Report”对话框，选



择“Computer”选项，如图 14-15 所示。

(2) 单击“Generate”按钮，在弹出的“Save File”对话框中输入报告的文件名“hardinfo_report”，选择文件的保存路径，如图 14-16 所示。

(3) 最后单击“保存”按钮，完成系统信息报告的生成操作。

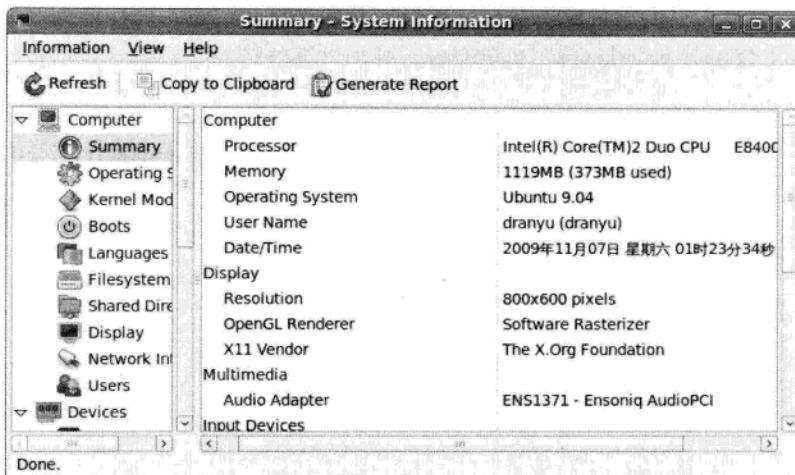


图 14-14 系统信息

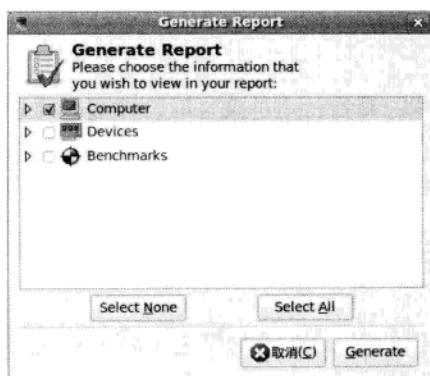


图 14-15 生成报告



图 14-16 保存报告文件

补充说明

在 hardinfo 应用程序中，系统管理员可查看系统的设备信息、系统中的用户信息、文件系统信息、网络连接信息和显示器设置信息等。hardinfo 里的信息显示也可使用 lspci 命令和 lshw 命令来显示。而网络的连接信息除了使用 hardinfo 应用程序查看，还可使用 ifconfig 命令来详细显示。



14.4.2 系统日志

系统日志是对特定事件的记录。一般的日志是按照一定的规范来编写的，是用户可直接阅读的文本文件。Ubuntu 系统有 3 个主要的日志子系统：连接时间日志、统计日志和错误日志。

Ubuntu 提供了图形查看系统日志的应用程序。Ubuntu 多数应用程序或进程的日志都会保存到/var/log 目录下。日志查看器也就默认读取此文件夹里的日志。选择顶部面板上“系统” | “系统管理” | “Log File Viewer” 菜单命令，弹出“系统日志查看器”对话框，如图 14-17 所示。



图 14-17 查看 dpkg 的日志

在系统日志查看器中查看系统日志操作方便、简单。查看所需的文件时，只需打开“系统日志查看器”对话框中左边的日志文件，则在查看器的右边显示日志的信息。

我们还可在系统日志查看器中打开其他日志文件，具体操作为：选择“文件” | “打开”菜单命令，在磁盘中找到需要查看的日志文件。

14.4.3 模块加载信息

Ubuntu 启动后会自动加载系统所需要的内核模块。Ubuntu 中的模块加载信息保存在 /proc/modules 文件中。我们可使用如下命令查看模块加载信息：

```
root@dranyu-desktop:/# cat /proc/modules
video 25360 0 - Live 0xf8401000
output 11008 1 video, Live 0xf83ee000
...
psmouse 61972 0 - Live 0xf823e000
pcspkr 10496 0 - Live 0xf8220000
serio_raw 13444 0 - Live 0xf8211000
snd_ens1371 30496 3 - Live 0xf81fd000
gameport 19468 1 snd_ens1371, Live 0xf81e2000
...
scsi_transport_spi 30080 1 mptspi, Live 0xf7d26000
...
softcursor 9984 1 bitblit, Live 0xf7c44000
```

我们还可使用 `lsmod` 命令来显示系统当前加载的所有模块，它的功能与 `cat /proc/modules` 相同。

在模块管理中，我们一定要认识的命令是 `modprobe`。使用 `modprobe` 命令可添加内核模块和删除不需要的模块。`modprobe` 命令的基本语法为：

```
modprobe [选项]
```

常用的 `modprobe` 选项说明如下。

- `-a: --all` //加载所有匹配模块。
- `-c: --showconfig` //显示当前使用的配置。
- `-d: --debug` //显示调试信息。
- `-h: --help` //帮助。
- `-k: --autoclean` //将指定模块设置为“自动清除”模式。
- Modules。
- `-l: --list` //显示所有匹配模块。
- `-n: --show` //仅仅显示要执行的操作,而不实际执行。
- `-q: --quiet` //不显示错误信息。
- `-r: --remove` //若在命令指定模块,则删除指定模块,否则,指定“自动清除”模式。
- `-s: --syslog` //将结果记录到系统记录中。
- `-t: --type moduletype` //指定模块类型。
- `-v: --verbose` //执行时显示详细信息。
- `-V: --version` //显示版本。
- `-C: --config configfile` //指定配置文件, 默认使用`/etc/modules.conf` 文件为配置文件。

例 14-27：添加内核模块

将 `xxx.ko` 模块添加到系统中，执行命令如下所示：

```
modprobe xxx.ko      #加载某个模块
```

例 14-28：卸载模块

```
modprobe -r xxx.ko    #卸载某个模块
```



14.5

系统备份与还原

对 Ubuntu 系统管理员来说，备份与恢复是日常管理工作中一个非常重要的环节。使用备份，我们可保存重要的数据或分区里的数据。当系统由于某种原因而崩溃的时候，我们可使用备份的数据还原系统。这样可降低系统崩溃带来的损害。

14.5.1 备份恢复的概念

备份是指将磁盘上的重要数据复制到其他的磁盘上。恢复是指在原文件出现错误和不正常时，将原有备份的数据再复制到原始目录，并用恢复后的数据启动服务或还原数据。

在为系统做备份时，我们应该先准备好备份的介质。目前，比较常用的备份介质有光盘、



磁带和硬盘。

14.5.2 备份恢复的策略

备份按照工作方式的不同可分为完全备份和增量备份。完全备份是指备份文件系统中的所有数据，而增量备份仅备份自上次备份以来改变了的数据。

- 完全备份

这种备份方式的好处在于：当发生数据丢失的时候，系统管理员可轻松地恢复丢失数据。其不足的地方是：造成硬件资源的大量浪费，再备份的时候耗时大。建议每周作一次完全备份，也要看具体的情况而定。

- 增量备份

增量备份只对改变的那部分数据进行备份。每次备份的数据量小，耗时小。这种备份方式的优点是节省了存储介质空间，缩短了备份时间。其不足之处是当发生数据丢失的时候，数据的恢复比较麻烦，且它的可靠性很差。

在现实操作中，这两种备份方式是结合在一起使用的。

在 Ubuntu 中并不是每个目录都需要我们去做数据备份。我们只对重要的数据和信息进行必要的备份。在 Ubuntu 中需要做备份的目录包括：/etc、/var、/home 和/root。建议不做备份的目录包括：/proc 和/dev。为了方便系统备份，一般在安装之前的磁盘分区步骤时就应该考虑为需要备份的目录独立创建一个分区。

14.5.3 tar 命令备份和还原/home 目录

tar 是 Linux 下最常用的备份和恢复文件工具，它最初是为了制作磁带档案而设计的，实际上用户可以用它将文件备份到任何介质上，包括软盘、硬盘和光盘等。使用 tar 命令做备份的优点包括：tar 命令的操作简单方便，其性能稳定，而且可靠性高。这种方式也存在不足之处，其中包括：不能备份特殊文件；只能制作完全备份，若用户要做增量备份，那就需要自己编写 shell 脚本。

在第 10 章中，我们已经讲述了 tar 的基本使用。在这里，我们来使用 tar 命令对/home 目录进行完全备份，具体的操作步骤如下所示：

(1) 选择磁盘的一个分区作为备份介质。使用 mount 命令将磁盘的第 7 个分区挂载到 /media/sda7 目录上。

(2) 在终端输入如下命令：

```
root@dranyu-desktop:~ # tar -cvpzf /media/sda7/backup.tgz /home
```

其中命令中的参数如下所示。

- C：新建一个备份文档。
- V：详细模式，tar 程序将在屏幕上实时输出所有信息。
- P：保存许可，并应用到所有文件。
- Z：采用‘gzip’压缩备份文件，以减小备份文件体积。
- F：说明备份文件存放的路径，/media/sda7/backup.tgz 是本例子中备份文件名。

经过上述两步就完成了/home 目录的备份工作。完成备份后，在我们需要恢复 home 的数

据时只要执行 tar 命令的反操作，执行命令如下所示：

```
tar -xvpzf /media/sda7/backup.tgz -C /home
```

其中参数 x 是告诉 tar 程序解压缩备份文件。-C 参数是指定 tar 程序解压缩到的目录。

补充说明

tar 命令也可实现增量式的备份策略。但在使用时，需要结合 find 命令。使用 find 命令找出需要备份的文件名列表，然后再使用 tar 命令进行备份。例如，需要备份过去 7 天更新的文件，可使用 find 命令先找出过去 7 天更新过的文件，然后使用 tar 的-T 参数指定需要备份的文件进行备份。代码如下所示：

```
Find /home -mtime -7 -print >/tmp/filelist
Tar -T /tmp/filelist -cf /tmp/backup.tar
```

14.5.4 dump 备份和 restore 还原文件

dump 是备份文件系统的实用工具，restore 命令是 dump 的反向操作，用于从 dump 备份文件中恢复整个文件系统或者某个文件。dump 和 restore 命令都能在网络上运行。因此，用户可以通过远程设备进行备份或恢复，但它们仅限于 ext2 文件系统。

1. dump 命令

dump 命令的基本语法格式如下所示：

```
dump [选项] 要备份的文件系统
```

其常用的选项如下所示。

- -0123456789：备份级别，其中 0 为完全备份，1~9 为增量备份。
- -b 块大小：指定块的大小，单位为 KB。
- -B 块数目：指定每个备份卷的块数目。
- -d 密度：设置磁带的密度，单位为 BPI。
- -f 文件名：指定存放备份的文件，可以是文件或物理设备。
- -T 日期：指定开始备份的时间。
- -u：备份完毕后，在/etc/dumpdates 中记录备份的文件系统、备份级别、日期与时间等。
- -w：与-W 类似，但仅显示需要备份的文件。
- -W：显示需要备份的文件及其最后一次备份的级别、时间与日期。

例 14-29：把文件系统备份到磁带

把/home 文件系统备份到第一个 SCSI 磁带驱动器上，命令如下所示：

```
root@dranyu-desktop:/etc# dump -0uf /dev/st0 /home
```

其中 0 表示完成完全备份，/home 是一个独立的文件系统，而/dev/st0 是 SCSI 磁带设备。

例 14-30：对/home 文件系统进行 3 级的增量式备份

将/home 文件系统中的文件使用 3 级的增量方式进行备份，备份到 backup 文件中，执行的命令如下所示：

```
root@dranyu-desktop:/etc# dump -3uf ~/backup /home
```

2. restore 命令

restore 命令是 dump 的反操作。使用 dump 命令进行备份的数据需要使用 restore 命令来



恢复。restore 命令可恢复全部文件系统、目录或一个 dump 归档文件中任意数量的文件。restore 命令的基本语法如下所示：

```
restore [选项] 备份的设备名或 dump 的归档文件
```

其中 restore 命令的常用选项如下。

- **-D** 文件系统：此选项允许用户指定文件系统的名称。
- **-f** 备份文件：此选项指定备份文件。
- **-i**：使用交互方式进行恢复操作。通过交互方式，用户可查找、添加、删除和抽取已经备份文件系统中的文件。
- **-r**：将整个备份档案恢复到当前目录中。
- **-s** 文件编号：当备份档案数据超过一卷磁带时，指定备份文件的编号。
- **-t**：列出备份档案中的文件。
- **-v**：显示 restore 命令的执行过程。
- **-x**：从备份档案中恢复指定文件。

例 14-31：恢复磁盘 st0 备份设备上的整个文件系统

在例 14-29 中，我们使用 dump 将/home 文件系统备份到了 st0 磁带上，现在使用 restore 命令恢复整个文件系统，执行命令如下所示：

```
root@dranyu-desktop:/home# restore -rf /dev/st0
```

若要恢复部分文件而不是整个文件系统，必须使用-x 选项，如例 14-32 所示。

例 14-32：从备份磁带中提取/home/dranyu 目录

/home 文件系统保存了 Ubuntu 多个用户的主文件夹，其中就包括了 dranyu 的目录，现在需要从 st0 备份设备中提取/home/dranyu 目录的内容，执行命令如下所示：

```
root@dranyu-desktop:/etc# restore -rf /dev/st0 /home/dranyu
```

14.6 小结



在 Ubuntu 中，系统的管理都可在图形界面中实现。本章讲述了系统管理中的用户和组管理、进程管理，以及查看系统的运行中的系统信息，在最后还讲述了系统管理员如何对系统中重要数据进行备份和恢复。

这一章对系统管理员来说尤其重要。通过这章的学习，可帮助系统管理员完成日常的系统维护和管理。

14.7 习题



1. 增加用户及删除用户使用什么命令？

2. /etc/passwd 文件内容和组织格式是什么？其每个字段有什么含义？

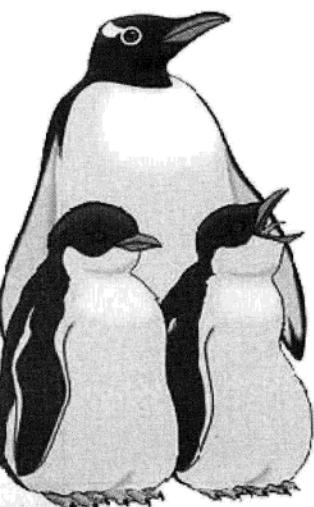
3. 如何封锁一个用户的账号？
4. 如何添加一个用户组？
5. /etc/group 文件内容和组织格式是什么？
6. 什么是进程？它与程序是什么关系？
7. 进程包括了哪几种状态？这些状态的转换关系是什么？
8. 如何查看系统的硬件信息？
9. Ubuntu 中有哪几个子日志系统？
10. 备份和恢复的概念是什么？备份有哪几种方式？
11. 如何使用 tar 命令对/etc 目录进行完全备份？
12. 使用 dump 命令将主文件夹备份到/usr/back 文件，然后使用 restore 命令将/usr/back 备份文件恢复到/home/sunny 目录。

Linux

第15章

网络管理

在现代计算机技术中，网络发挥着越来越重要的作用，如今大部分的计算机主要用途都是用来浏览 Web 站点或者收发电子邮件，而 Linux 也正是随着网络的发展而逐步发展完善的，Ubuntu 虽然发展的时间不长，但是同样继承了 Linux 强大的网络功能，它只有在网络中才能发挥它的全部功能。





15.1 Linux 网络基础

若想成为专业的网络管理员，就必须熟练掌握 Linux 操作系统的管理与开发。任何一个专业的网络安全工具和系统都不可能百分之百满足日益变化和更新的网络管理需求。因此，掌握有效管理网络系统各种软硬件资源的方法是优秀网络管理员的首要任务。

15.1.1 计算机网络体系基本概念

简单地说，只要是两台以上的计算机，通过网线或者是其他的网络媒体（如无线网卡、光纤专线等）连接在一起，以进行文件传输和消息传递功能，那就形成一个“网络世界”。

1. 网络类型

网络类型的划分标准各种各样，但是从地理范围划分是一种大家都认可的通用网络划分标准。网络类型划分并没有严格意义上地理范围的区分，只能是一个定性的概念。下面简要介绍这几种计算机网络。

(1) 局域网 (Local Area Network, LAN)

通常我们常见的“LAN”就是指局域网，这是我们最常见、应用最广的一种网络。所谓局域网，就是在局部地区范围内的网络，它所覆盖的地区范围较小。局域网在计算机数量配置上没有太多的限制，少的可以只有两台，多的可达几百台。一般来说，在企业局域网中，工作站的数量在几十到两百台之间。在网络所涉及的地理距离上来说，可以是几米至几十千米以内。局域网一般位于一个建筑物或一个单位内，不存在循径问题，不包括网络层的应用。

这种网络的特点就是：连接范围窄、用户数少、配置容易、连接速率高。目前，局域网最快的速率要算现今的 10Gbps 以太网。IEEE 的 802 标准委员会定义了多种主要的 LAN 网：以太网 (Ethernet)、令牌环网 (Token Ring)、光纤分布式接口网络 (FDDI)、异步传输模式网 (ATM)，以及最新的无线局域网 (WLAN)。这些都将在后面详细介绍。

(2) 城域网 (Metropolitan Area Network, MAN)

这种网络一般来说是在一个城市，但不在同一地理小区范围内的计算机互连。这种网络的连接距离可以在 10~100 千米，它采用的是 IEEE802.6 标准。MAN 与 LAN 相比扩展的距离更长，连接的计算机数量更多，在地理范围上可以说是 LAN 网络的延伸。在一个大型城市或都市地区，一个 MAN 网络通常连接着多个 LAN 网。如连接政府机构的 LAN、医院的 LAN、电信的 LAN、公司企业的 LAN 等。由于光纤连接的引入，使 MAN 中高速的 LAN 互连成为可能。

(3) 广域网 (Wide Area Network, WAN)

这种网络也称为远程网，所覆盖的范围比城域网 (MAN) 更广，它一般是在不同城市之间的 LAN 或者 MAN 网络互连，地理范围可从几百千米到几千千米。因为距离较远，信息衰减比较严重，所以这种网络一般是要租用专线，通过 IMP (接口信息处理) 协议和线路连接起来，构成网状结构，解决循径问题。这种城域网因为所连接的用户多，总出口带宽有限，所以用户的终端连接速率一般较低，通常为 9.6Kbps~45Mbps 如邮电部的 CHINANET、



CHINAPAC 和 CHINADDN 网。

(4) 互连网 (Internet)

互连网因其英文单词“Internet”的谐音，又称为“因特网”。在互连网应用如此发达的今天，它已是我们每天都要打交道的一种网络，无论从地理范围，还是从网络规模来讲它都是最大的一种网络，就是我们常说的“Web”、“WWW”和“万维网”等多种叫法。从地理范围来说，它可以是全球计算机的互连，这种网络的最大的特点就是不定性，整个网络的计算机每时每刻随着人们网络的接入在不断地变化。当您连在互连网上的时候，您的计算机可以算是互连网的一部分，但一旦当您断开互连网的连接时，您的计算机就不属于互连网了。但它的优点也是非常明显的，就是信息量大，传播广，无论你身处何地，只要连上互连网你就可以对任何可以连网用户发出你的信函和广告。因为这种网络的复杂性，所以这种网络实现的技术也是非常复杂的，这一点我们可以通过后面要讲的几种互连网接入设备详细地了解到。

上面讲了网络的几种分类，其实在现实生活中我们真正遇得最多的还要算是局域网，因为它可大可小，无论在单位还是在家庭实现起来都比较容易，应用也是最广泛的一种网络。所以，在下面有必要对局域网及局域网中的接入设备作进一步的认识。

2. 网络体系结构

计算机网络体系结构，可以定义为是网络协议的层次划分与各层协议的集合，图 15-1 给出了网络体系结构中协议、层、服务与接口之间的关系。

在图 15-1 中，同一层的协议根据该层所要实现的功能来确定。各对等层之间的协议功能由相应的底层提供服务完成。

层次化的网络体系的优点在于每层实现相对独立的功能，层与层之间通过接口来提供服务，每一层都对上层屏蔽如何实现协议的具体细节，使网络体系结构做到与具体物理实现无关。层次结构允许连接到网络的主机和终端型号、性能可以不同，但只要遵守相同的协议即可以实现相互操作。高层用户可以从具有相同功能的协议、层开始进行互连，使网络成为开放式系统。这里“开放”指的是按照相同协议任意两系统之间可以进行通信。因此，层次结构便于系统的实现和便于系统的维护。

对于不同系统实体间互连互操作这样一个复杂的工程设计问题，如果不采用分层次分解处理，则会产生由于任何错误或性能修改而影响整体设计的弊端。

相邻协议层之间的接口包括两相邻协议层之间所有调用和服务的集合，服务是第 i 层向相邻高层提供服务，调用是相邻高层通过原语或过程调用相邻低层的服务。

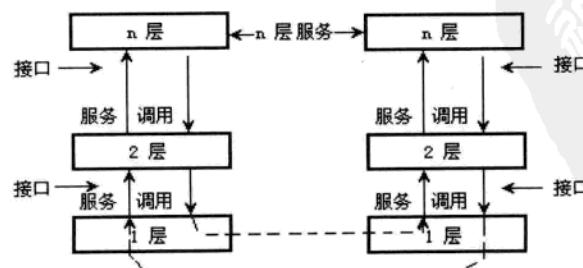


图 15-1 计算机网络层次结构

对等层之间进行通信时，数据传送方式并不是由第 i 层发方直接发送到第 i 层收方。而是每一层都把数据和控制信息组成的报文分组传输到它的相邻低层，直到物理传输介质。接收时，则是每一层从它的相邻低层接收相应的分组数据，在去掉与本层有关的控制信息后，将有效数据传送给其相邻上层。

国际标准化组织 ISO (International Standards Organization) 在 80 年代提出的开放系统互连参考模型 OSI (Open System Interconnection)，这个模型将计算机网络通信协议分为七层。

OSI 网络体系结构如图 15-2 所示。在 OSI 网络体系结构中除了物理层之外，网络中数据的实际传输方向是垂直的。数据由用户发送进程发送给应用层，向下经表示层、会话层等到达物理层，再经传输媒体传到接收端，由接收端物理层接收，向上经数据链路层等到达应用层，再由用户获取。数据在由发送进程交给应用层时，由应用层加上该层有关控制和识别信息，再向下传送，这一过程一直重复到物理层。在接收端信息向上传递时，各层的有关控制和识别信息被逐层剥去，最后数据送到接收进程。

现在一般在制定网络协议和标准时，都把 ISO/OSI 参考模型作为参照基准，并说明与该参照基准的对应关系。例如，在 IEEE802 局域网 LAN 标准中，只定义了物理层和数据链路层，并且增强了数据链路层的功能。在广域网 WAN 协议中，CCITT 的 X.25 建议包含了物理层、数据链路层和网络层 3 层协议。一般来说，网络的低层协议决定了一个网络系统的传输特性，例如所采用的传输介质、拓扑结构及介质访问控制方法等，这些通常由硬件来实现；网络的高层协议则提供了与网络硬件结构无关的，更加完善的网络服务和应用环境，这些通常是由网络操作系统来实现的。



图 15-2 ISO/OSI 参考模型

15.1.2 TCP/IP 协议族

TCP/IP 也称“国际协议簇”，即不仅指 TCP/IP 协议本身，而且包括与其有关的协议。TCP 为传输控制协议，IP 为网际协议，是网络层最重要的协议。采用 TCP/IP 协议通过互连网传送信息可减少网络中的传输阻塞，方便大批量的数据在网上传输，从而提高网络的传输效率。TCP/IP 协议族中包括上百个互为关连的协议，其中有：Telnet (Remote Login) 提供远程登录功能；FTP (File Transfer Protocol) 远程文件传输协议，允许用户将远程主机上的文件复制到自己的计算机上；SMTP (Simple Message Transfer Protocol) 是简单信息传输协议，主要用于传输电子邮件；NFS (Network File Server) 是网络文件服务器，可使多台计算机透



明地访问彼此的目录；UDP（User Datagram Protocol）是一个用户数据包协议。

Internet 网络体系结构以 TCP/IP 协议为核心。其中 IP 协议用于为各种不同的通信子网或局域网提供统一的互连平台，TCP 协议则用于为应用程序提供端到端的控制和通信功能。

1. TCP/IP 协议层次和协议集

由于 TCP/IP 协议开发先于 OSI 参考模型，故不完全和 OSI 参考模型对应。从体系结构上来看，TCP/IP 基本上是 OSI 的 7 层模型的简化，它只分为 4 层：网络接口层、网络层、传输层和应用层。

（1）网络接口层（SubNetwork Layer）

TCP/IP 协议的网络接口层与 OSI 协议的物理层、数据链路层，以及网络层的部分相对应。该层没有规定新的物理层和数据链路层协议，允许通信子网采用已有的或将来的各种协议，例如以太网的 802.3 协议，分组交换网的 X.25 协议等。该层只定义了 TCP/IP 与各种通信子网之间的网络接口。网络接口层的功能是传输经网络层处理过的消息。

（2）网络层（Internet Layer）

该层与 OSI 网络层相对应，由于它是针对网际环境设计的，具有更强的网际通信能力。网络层协议为 IP 协议。它将传输层送来的消息组装成 IP 数据包，并且把 IP 数据包传递给网络接口层。IP 提供端到端分组发送功能，标识网络号及主机节点地址的功能，为使 IP 数据包长度与通信子网允许的数据包长度相匹配，提供了数据分段和重新组装的功能。

该层还提供建立独立的局域网之间的互连网络。在互连网络中，连接两个以上网络的节点称为路由器(网关)，其允许网间的报文根据它的目的地址通过路由器传送到另一个网络。

（3）传输层(Transport Layer)

该层与 OSI 传输层相对应，为应用程序提供端到端通信功能。传输层有 3 个主要协议，其分别为传输控制协议 TCP，用户数据报协议 UDP 和互连网控制消息协议 ICMP。

TCP 协议负责将用户数据按规定长度组成数据包发送，在接收端对数据包按顺序进行分解重组以恢复用户数据。TCP 协议是以建立高可靠性信息传输为目的，为了可靠传输数据，该协议具有数据包的顺序控制、差错检测、检验，以及再发送控制等功能。

UDP 协议负责主机和网关，以及 Internet 运行管理中心等的信息通信，控制管理网络的运行。

ICMP 协议负责当数据包传输有误时，发送出错信息给数据包发送端主机，另外还具有控制数据包流量的功能。

（4）应用层（Application Layer）

该层包含了 OSI 会话层，表示层和应用层的功能，为用户提供各类服务。例如，远程登录、文件传输、电子邮件、Web 服务器等。

TCP/IP 协议集，以及各层次与 OSI 各层次对应关系如图 15-3 所示。

OSI 层次		TCP/IP 层次		TCP/IP 协议集			
5~7 层		应用层		SMTP	DNS	FTP	RPC
4 层		传输层		TCP			SNMP
3 层		网络层		IP (ICMP, ARP, RARP)			
1~2 层		网络接口层		Ethernet	Token-Ring	100BASE-T	Others

图 15-3 TCP/IP 协议集以及分层结构

2. TCP / IP 分层工作原理

(1) TCP / IP 分层

TCP / IP 的分层工作原理如图 15-4 所示, 表示了两台主机上的应用程序之间传输报文的路径。主机 B 上的第 n 层接收到的正是主机 A 上的第 n 层发送出来的对象。

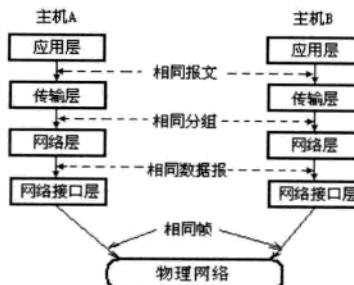


图 15-4 TCP/IP 分层工作原理

(2) TCP/IP 模型的分界线

TCP/IP 协议的概念性层次包含两个重要的划分, 一个是协议地址的划分界线, 以区分不同的循址, 另一个是操作系统的划分, 以区分系统程序与应用程序。

网络接口层处理物理地址, 该层协议已集成到操作系统中, 属于系统软件。网络层与传输层处理 IP 地址, 这两层协议也已集成到操作系统中, 属于系统软件。应用层协议未集成到操作系统中, 属于非系统软件。

(3) 网络协议与操作系统的关系

计算机体系结构中的低层部分, 即物理层和数据链路层协议, 是和通信设备等硬件密切相关的, 需要由网络接口卡和网络互连设备来实现。计算机体系结构中的高层部分, 网络层协议和传输层协议, 由计算机主机操作系统来实现, 作为核心线程方式运行。网络层与数据链路层协议的接口, 以及服务调用, 是通过操作系统中网卡驱动程序来实现的。处于 TCP 之上的应用层协议大部分使用 TCP/IP 协议提供的系统调用, 它们处于用户程序的地位, 以用户线程方式执行。

15.1.3 TCP 协议

TCP 为传输层协议, 提供可靠的进程间通信机制, 该协议可以自动纠正各种差错, TCP 支持许多高层协议 ULP (Upper Level Protocol), 它提供一对 ULP 之间面向连接的传输服务, 包括建立连接、数据传输和终止连接。TCP 主要通过套接字 (Socket) 为 ULP 提供面向连接的传输服务。

1. 接口和套接字

传输层协议和应用程序之间的数据接口由运行在某主机上的 TCP 协议为其分配端口号。为了使其具有全局唯一性, 将此端口号和 IP 地址合起来组成套接字。

使用 TCP 协议的网络应用程序分为服务程序和客户程序。服务程序为其他主机提供服



务，服务程序使用固定的端口号，例如 Telnet 服务端口号为 23，FTP 为 21 号，E-mail 为 25 号。客户程序使用服务程序提供的服务，客户程序可选择通信端口号。服务程序运行后，在各自端口等待，如果希望使用服务程序提供的服务，可向该服务对应的套接字上发送消息。

2. TCP 报文格式

TCP 协议中的基本传输单元为段，TCP 报文又称为 TCP 段。一个 TCP 段由段头和数据流两部分组成。其格式如图 15-5 所示，说明见表 15-1。

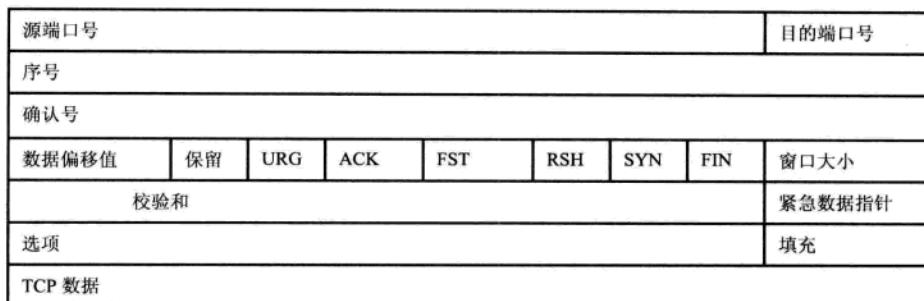


图 15-5 TCP 报文格式

表 15-1 TCP 报文格式说明

报头字段名	位 数	说 明
源端口号	16	本地通信端口
目的端口号	16	远地通信端口
序号(SEQ)	32	数据部分第一个字节的序列号
确认号(ACK)	32	表示本地希望接收的下一个数据字节的序号
数据偏移值	4	该 TCP 段中数据的起始位置
URG	1	紧急数据指针有效标志，指示本段中包含紧急数据
ACK	1	确认标志，指示段中确认号有效
PST	1	复位连接标志，指示本段为复位段
RSH	1	PUSH 操作标志
SYN	1	建立同步连接标志
FIN	1	本地数据发送结束，终止连接标志
窗口大小	16	本地接收窗口（接收缓冲区）大小
校验和	16	包括 TCP 报头和数据在内的校验和
紧急数据指针	16	指示从发送数据序列号开始到紧急数据之后的第一个字节偏移值
选项	可变	提供选服务
填充	可变	保证 TCP 段头以 32 位为边界对齐

TCP 数据流是无结构的字节流，这一特征允许 TCP 段长可变。两个使用 TCP 协议进行通信的对等实体间的通信，一般要经历建立连接、数据传输和终止连接 3 个阶段。TCP 协议

通过状态转换机制，可以保证可靠地执行各个阶段，为上层应用提供双向、可靠、顺序及无重复的数据流传输服务。

3. 建立连接

TCP 的连接建立，需要发送序列号和接收序列号，初始序列号由 TCP 在创建连接时分配，建立连接实际上是双方序列号同步的过程，建立连接时，每一方都发送自己的初始序列号，并且接收对方发送的序列号作为接收序列号，通过“三次握手”机制建立连接。TCP 协议可以处理一方发起的连接请求或由双方同时发起的连接请求。正常连接建立过程如图 8-14 所示，同步标志 SYN 用于连接建立阶段。由发起方发送初始序列号为 X，且 SYN 标志置位的段；由接收方回送初始序列为 Y、SYN 置位、ACK 标志置位且应答序列号为 X + 1（表示希望接收的下一个数据字节的序列号）的响应段；最后发起方发送以 ACK 置位，应答序列号为 Y + 1 应答的段，建立起连接。

4. 关闭连接

连接的关闭由双方共同完成。关闭请求可以由一方发起也可以由双方同时发起。请求关闭方向对方发送 FIN 段请求关闭连接，接收方必须使用 ACK 段给予确认。关闭过程也是一个“三次握手”过程。

15.1.4 UDP 协议

UDP(User Datagram Protocol) 用户数据包协议是 OSI 参考模型中一种无连接的传输层协议，提供面向事务的简单不可靠信息传送服务。是一个简单的面向数据包的传输层协议，IETF RFC 768 是 UDP 的正式规范。UDP 协议基本上是 IP 协议与上层协议的接口。UDP 协议适用端口分别运行在同一台设备上的多个应用程序。

与 TCP 不同，UDP 并不提供对 IP 协议的可靠机制、流控制，以及错误恢复功能等。由于 UDP 比较简单，UDP 头包含很少的字节，比 TCP 负载消耗少。UDP 协议具有如下的特性：

(1) UDP 是一个无连接协议，传输数据之前源端和终端不建立连接，当它想传送时就简单地去抓取来自应用程序的数据，并尽可能快地把它应用到网络上。在发送端，UDP 传送数据的速度仅仅是受应用程序生成数据的速度、计算机的能力和传输带宽的限制；在接收端，UDP 把每个消息段放在队列中，应用程序每次从队列中读一个消息段。

(2) 由于传输数据不建立连接，因此也就不需要维护连接状态，包括收发状态等，因此一台服务器可同时向多个客户机传输相同的消息。

(3) UDP 信息包的标题很短，只有 8 个字节，相对于 TCP 的 20 个字节信息包的额外开销很小。

(4) 吞吐量不受拥挤控制算法的调节，只受应用软件生成数据的速率、传输带宽、源端和终端主机性能的限制。

显然 UDP 是一个不可靠的协议，那么还有什么使用价值或必要呢？其实不然，在有些情况下，UDP 协议可能会变得非常有用。因为 UDP 具有 TCP 所望尘莫及的速度优势。虽然 TCP 协议中植入了各种安全保障功能，但在实际执行的过程中会占用大量的系统开销，无疑使速度受到严重的影响。反观 UDP 由于排除了信息可靠传递机制，将安全和排序等功能移交给上



层应用来完成，极大地降低了执行时间，使速度得到了保证。

UDP 适用于不需要 TCP 可靠机制的情形，比如，当高层协议或应用程序提供错误和流控制功能的时候。UDP 是传输层协议，服务于很多知名应用层协议，包括网络文件系统（NFS）、简单网络管理协议（SNMP）、域名系统（DNS），以及简单文件传输系统（TFTP）、动态主机配置协议（DHCP）、路由信息协议（RIP）和某些影音串流服务等。

关于 UDP 协议的最早规范是 RFC768，于 1980 年发布。尽管时间已经很长，但是 UDP 协议仍然继续在主流应用中发挥作用。包括视频电话会议系统在内的许多应用都证明了 UDP 协议的存在价值。因为相对于可靠性来说，这些应用更加注重实际性能，所以为了获得更好的使用效果（例如，更高的画面帧刷新速率）往往可以牺牲一定的可靠性（例如，会面质量）。这就是 UDP 和 TCP 两种协议的权衡之处。根据不同的环境和特点，两种传输协议都将在今后的网络世界中发挥更加重要的作用。

15.1.5 IP 协议

IP 协议为网络层协议，其内容包括：基本传输单元，也就是 IP 报文的类型与定义、IP 报文的地址，以及分配方法、IP 报文的路由转发，以及 IP 报文的分段与重组。

IP 协议仅具有有限的报错功能，数据包在传输过程中的差错检测和恢复都由传输层协议 TCP 来完成。

1. IP 数据报

IP 数据报包括报文头和正文部分，其格式如图 15-6 所示，以及说明见表 15-2。



图 15-6 IP 协议数据报格式

表 15-2 IP 数据报格式说明

名 字	位 数	说 明
版本号	4	此协议版本号
IP 报头长度	4	IP 数据报的报头长度（32 位）
服务级别	8	规定优先级、传送速率、吞吐量和可靠性等参数
报文长度	16	报头和数据两部分总长度（字节）
标识	16	数据报标识，便于数据报分段与重装

续表

名 字	位 数	说 明
标志	3	数据报是否分段标志
分段偏移	13	以 64 位为单位表示的分段偏移
生存时间	8	允许数据报在网间传输的存活时间
用户协议号	8	指出发送数据报的上层协议
报头校验和	16	只用于报头检查
源 IP 地址	32	数据报的源主机 IP 地址
目的 IP 地址	32	数据报的目的主机 IP 地址
任选项	可变	规定附加服务，如时间戳、错误报告及特殊路由等
填充	可变	保证 IP 报头以 32 位为边界对齐

2. 数据报分段与重装

在各种物理网络中，数据帧的长度可以不一样，在互连网络中，IP 协议首先要根据物理网络所允许的最大发送长度对上层协议提交的数据报进行长度检查，需要时把数据报分成若干个段发送，对数据报分段后，形成多个 IP 数据报，为了说明被分段数据报和 IP 数据报的关系，被分段 IP 数据报需设置相同的标识（IP）。每一个被分段的 IP 数据报要重新计算其报文长度，并且要表明该 IP 数据报在原始数据报中的位置。

在互连网络中，被分段的数据报独立传输，在经路由器转发时，由于路由可能不同，导致到达目标主机时，接收到的 IP 数据报顺序与发送时的顺序不一致。因此，必须根据 IP 数据报中相关字段（标识、长度、偏移及标志等）将被分段的数据报重新组装恢复为原始 IP 数据报（保留第一段 IP 数据报头），根据用户协议号提交给上层协议。

3. IP 地址

IP 地址是 IP 协议的核心，IP 地址格式如图 15-7 所示。



图 15-7 IP 地址格式

A 类地址中网络地址 126 个（1 个字节），主机地址 16387064 个（3 个字节），其范围为 0.0.0~127.255.255.255。A 类地址用于大型网络，由国际网络中心来分配。



B 类地址中网络地址 16256 个（2 个字节），主机地址 64576 个（2 个字节），其范围为 128.0.0.0 ~ 191.255.255.255。B 类地址用于各地区的网管中心，由欧洲、北美和亚太三大区域网络信息中心负责分配。

C 类地址中网络地址 2064512 个（3 个字节），主机地址 254 个（1 个字节），其范围为 192.0.0.0 ~ 223.255.255.255。C 类地址用于校园网或企业网。由国家或地区网络信息中心负责分配。

D 类地址为组播地址，E 类地址保留。

4. 子网和 IP 地址屏蔽码

由于 Internet 发展迅速，网络地址成为宝贵资源，为了提高 IP 地址的利用率，将大型网络划分为若干个逻辑上相互独立的子网，但网络地址不变，原主机地址成为子网主机地址。

IP 地址屏蔽码是和 IP 地址配合的，用于说明某大型网络所具有的子网数，以及每个子网所包含的主机数，屏蔽码是对地址分类的扩展。

IP 协议为每个网络接口分配一个 IP 地址，在有子网的 IP 地址中，其子网号是通过主机号字段的高几位二进制位来表示的，所占位数与子网数对应。如果把该接口 IP 地址与其屏蔽码相与，屏蔽码将 IP 地址中主机字段主机号屏蔽掉，即可得到该接口所在网络的子网号。以 C 类地址为例，例如，屏蔽码 255.255.255.192（11000000B）表示有 4 个子网号，每个子网可以有 64 台主机，设 IP 地址为：

202.117.80.1 主机号为 00000001

202.117.80.3 主机号为 00000011

202.117.80.130 主机号为 10000010

将上述 IP 地址与屏蔽码相与，可看出前两个 IP 地址的子网号为 0 号，属于同一网段，而第三个 IP 地址的子网号为 1 号，不属于同一网段。

15.2 网络工具 GNOME-nettool



Ubuntu 中集成了图形的网络管理工具 GNOME-nettool 应用程序，网络管理员可通过使用此应用程序查看本机网络设备的连接信息、验证本机与远程计算机的连接、查看从本机出发到指定远程计算机所经过的路由信息和扫描开放的端口信息。

选择“系统”|“系统管理”|“网络工具”菜单命令，弹出“设备-网络工具”对话框如图 15-8 所示。

15.2.1 查看本地 IP 地址

使用网络工具查看本地 IP 地址的方法为：在“网络工具”对话框中切换到“设备”标签，然后在“网络设备”下拉列表中选择“以太网卡（eth4）”选项，本机中 eth4 设备所获取的 IP 地址信息显示如图 15-9 所示。



图 15-8 网络工具



图 15-9 IP 地址的查看

补充说明

ethx 是 Linux 表示以太网卡设备的格式，其中 x 是一个数字。

15.2.2 验证与远程计算机的连接

作为网络管理员，可使用网络工具来验证本地计算机是否可连接到远程计算机，操作方法为：在网络工具对话框中切换至“Ping”标签页，在网络地址输入栏上输入远程的网址或 IP 地址，如 www.google.com，然后单击“Ping”按钮，显示数据包在本地计算机和远程计算机 www.google.com 之间传输的状态信息，如图 15-10 所示。在每个色块下显示了每个序列的数据包发送和回收所需的时间，如序列号为 1 的数据包发送和回收所发的时间为 8.44 毫秒。“成功传送的包数：100%”的数据表示本地计算机与远程计算机连接正常。

15.2.3 查看路由表信息

路由器（网关）的主要工作就是为经过路由器的每个数据包寻找一条最佳传输路径，并将该数据有效地传送到目的站点。由此可见，选择最佳路径的策略即路由算法是路由器的关键所在。为了完成这项工作，在路由器中保存着各种传输路径的相关数据——路由表，供路由选择时使用。路由表可以是由系统管理员固定设置好的，也可以由系统动态修改，可以由路由器自动调整，也可以由主机控制。

系统管理员需要了解路由表的信息，才可正确地分析网络的连接状态，进而管理网络设备与互连网的连接。使用网络工具查看路由表信息的方法为：在网络工具对话框中切换至“网络统计”标签页，然后在“显示”选项框中选择“路由表信息”选项，单击“网络统计”按钮，在路由信息列表中显示了本地计算机的路由表信息，如图 15-11 所示。

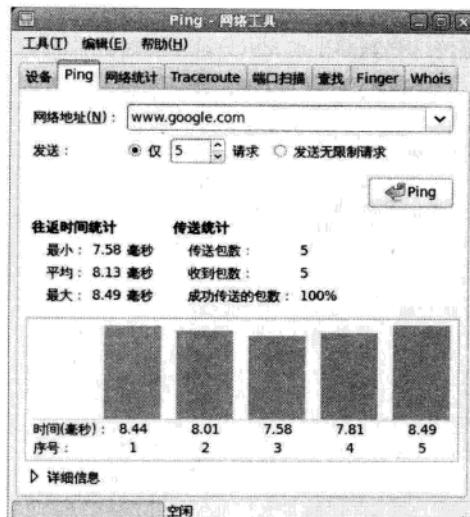


图 15-10 验证与远程计算机的连接

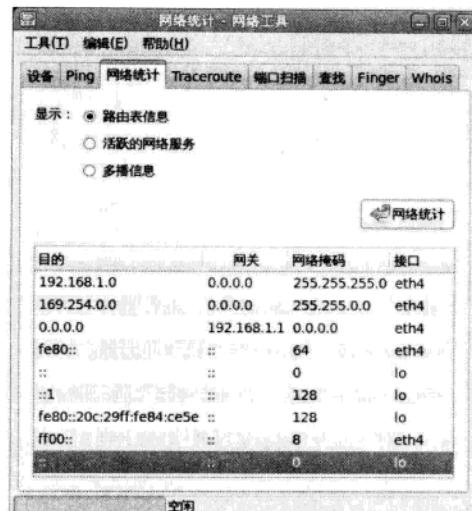


图 15-11 路由表信息

15.2.4 扫描开放的端口

把好计算机与互连网的关闭端口，可保证计算机的安全。需要知道本地计算机的端口开放情况可使用网络工具的“端口扫描”功能，具体操作方法为：在网络工具对话框中切换至“端口扫描”标签页，在“网络地址”栏输入需要进行端口扫描的域名或 IP 地址，如“192.168.1.104”，然后单击“扫描”，则 192.168.1.104 主机上的开放端口就会显示在结果显示列表中，如图 15-12 所示。

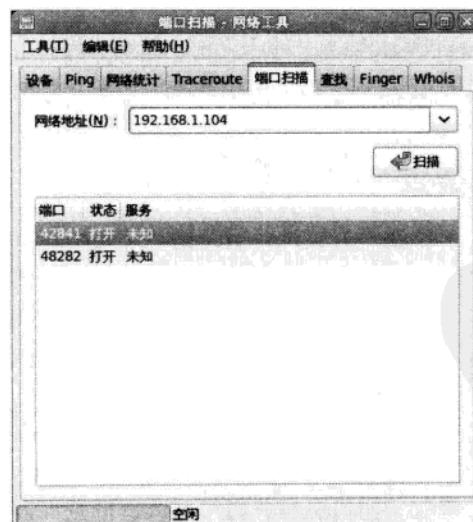


图 15-12 端口扫描



15.3 网络配置

Ubuntu 提供了一个图形界面网络配置工具—nm-connection-editor，使用此工具可配置网络连接的方式。网络管理员除了掌握使用方便的图形界面的配置工具，还必须了解和掌握 Ubuntu 中网络配置的网络命令。

补充说明

使用网络配置工具配置静态的 IP 地址和配置 DSL 的网络连接的方法在第 6.10 节上已讲述。

15.3.1 网络配置文件

对于网络管理员来说，了解网络配置文件的作用很有必要。

1. 主机地址设置文件——/etc/hosts

主机地址设置文件可以配置主机 Ip 和主机名的对应关系，对于服务器类型的 Linux 系统其作用还是不可忽略的。在局域网或是 Internet 上，每台主机都有一个 IP 地址，它区分开每台主机，并可以根据 Ip 进行通信。但 Ip 地址不方便记忆，所以出现了域名。在一个局域网中，每台机器都有一个主机名，用于区分主机，便于相互访问。

此文件告诉本主机哪些域名对应哪些 Ip，哪些主机名对应哪些 Ip，比如文件中有这样的定义：

```
127.0.0.1 dranyu-desktop localhost.localdomain localhost
```

说明本主机的主机名为 dranyu-desktop，而 localhost.localdomain 和 localhost 是本主机的别名。此文件的默认内容如下：

```
127.0.0.1 dranyu-desktop localhost.localdomain localhost
127.0.1.1 dranyu-desktop
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

一般情况下，hosts 文件的每行为一个主机，每行由 3 部分组成，每个部分由空格隔开。其中#号开头的行作说明，不被系统解释。这 3 部分包括：网络 IP 地址、主机名或域名和主机名别名。然而每行也可以是两部分，即包含主机 IP 地址和主机名，如默认文件内容里的“127.0.0.1 dranyu-desktop”行。Ubuntu 的默认主机地址配置文件还包括了 IP 为 IPv6 格式的主机标识行，如“::1 localhost ip6-localhost ip6-loopback”。

补充说明

解释一下主机名（hostname）和域名（Domain）的区别：主机名通常在局域网内使用，通过 hosts 文件，主机名就被解析到对应 Ip；域名通常在 Internet 上使用，但如果本机不想使用 Internet 上的域名解析，这时就可以更改 hosts 文件，加入自己的域名解析。



2. 网络服务数据文件——/etc/services

/etc/services 配置文件中保存了服务名和端口号之间的映射，不少系统程序要使用这个文件。此文件中包含 Internet 服务名称、使用的连接端口号与所使用的网络协议等信息。例如网络的 Web 服务器的默认端口是 80，我们使用如下代码查看/etc/services 文件中 Web 服务器与端口号的映射关系：

```
dranyu@dranyu-desktop:/etc/network$ cat /etc/services | grep 80 //显示端口号与 80 匹配的记录
www      80/tcp      http      # WorldWideWeb HTTP
www      80/udp      # HyperText Transfer Protocol
socks    1080/tcp    # socks proxy server
socks    1080/udp
http-alt 8080/tcp webcache # WWW caching service
http-alt 8080/udp      # WWW caching service
amanda   10080/tcp    # amanda backup services
amanda   10080/udp
omirr    808/tcp      omirrd    # online mirror
omirr    808/udp      omirrd
canna    5680/tcp    # cannaserver
zope-ftp 8021/tcp    # zope management by ftp
tproxy   8081/tcp    # Transparent Proxy
omniORB  8088/tcp    # OmniORB
omniORB  8088/udp
```

在结果的第一行中 WWW 是网络服务的名称，80 指定端口号，tcp 表示网络协议，而“#”后边的字符串是备注。基本上每个服务都必须使用唯一的连接端口号/通信协议对应，因而若两个服务需使用同一个端口号，那么这两个服务必须使用不同的通信协议；若两个服务须使用同一个通信协议，则一定需要使用不同的端口号。上述例子中，第一行的服务和第二行的服务使用了同一个端口号，可是使用了不同的通信协议，第一行的服务使用了 tcp 通信协议，而第二行服务使用了 udp 通信协议。

Ubuntu 中端口号的范围是有限的，其范围为 0~65535，其中可分为 3 类：0~1023 是专供服务器使用的；1024~49151 是供所有 Internet 上登录的连接所使用的；而 49152~65535 是其他动态或私人设定的连接端口号。有些端口号是指定的，如 20 是 ftp 服务器的数据接收端口号；3306 是 Mysql 数据库服务器的默认端口号。

3. 解析主机名的设置文件——/etc/host.conf

/etc/host.conf 配置文件指定以何种方式解析主机名，Linux 通过解析器来获得主机名对应的 IP 地址。此文件指定了计算机解析主机名时搜索解析数据的顺序，设置是否允许指定的主机有多个地址。在 Ubuntu 中，默认的/etc/host.conf 文件的内容如下所示：

```
# The "order" line is only used by old versions of the C library.
order hosts,bind
multi on
```

在/etc/host.conf 文件中，每行表示一项功能。上例中 order 行指定主机名查询顺序，其参数为用逗号隔开的查找方法，支持的查找方法包括 bind、hosts 和 nis，分别代表 DNS、/etc/hosts 和 NIS，此处默认配置文件设置了先查询 “/etc/hosts” 文件然后再使用 DNS 来解析域名。而 “multi on” 行表示允许 “/etc/hosts” 文件中指定的主机可拥有多个地址。拥有多个 IP 地址的主机一般称为具有多个网络界面。

除了默认设定的几个功能，/etc/host.conf 还可指定如下功能行：

- trim

此项表明当通过 DNS 进行地址到主机名的转换时，域名将从主机名中被裁剪掉，trim 可以被多个域包含多次，但对/etc/hosts 和 NIS 查询方法不起作用。

- nospoof

此项表示是否允许对该服务器进行 IP 地址欺骗，值为 on 表示不允许，IP 欺骗是一种攻击系统安全的手段，通过把 IP 地址伪装成别的计算机，来取得其他计算机的信任。

- alert

在设置了 nospoof 指令为 on 时，alert 控制着是否使用 syslog 工具对欺骗的企图进行记录，值为 on 表示使用，off 表示不使用，默认值是 off。

- recorder

如果被设置为 on，所有的查询将被重新排序，所以在同一子网中的主机将首选被返回，默认值为 off。

4. 网络接口参数配置文件——/etc/network/interfaces

Ubuntu 的网络接口参数配置文件/etc/network/interfaces 指定网卡接口的连接方式，此文件的默认内容如下所示：

```
auto lo
iface lo inet loopback
```

其中，第一行表示 lo 接口会在系统启动时被自动配置；第二行表示将 lo 接口设置为一个本地回环（loopback）地址。

网络管理员可在此文件中设置网络连接的方式，如为网络分配一个静态的 IP 地址，可将 /etc/network/interfaces 文件的内容修改为如下所示代码：

```
auto eth0
iface eth0 inet static
address 192.168.1.112
network 192.168.1.0
netmask 255.255.255.0
gateway 192.168.1.1
```

其中第一行 auto eth0 表示在系统启动时启动 eth0 网络设备；第二行表示使用静态分配 IP 的方式获取 IP 连接地址；address 行指定了本机的静态 IP 地址；network 行指定了所在的网络；netmask 行表示网络掩码；gateway 行表示网关。

5. DNS 客户端配置文件——/etc/resolv.conf

DNS 客户端配置文件/etc/resolv.conf 包含了主机的域名搜索顺序和 DNS 服务器的地址，每一行应包含一个关键字和一个或多个的由空格隔开的参数。此文件的内容由网络管理工具自动修改，也可在系统启动之后进行人为修改。将网络连接方式设置为自动获取网络的 IP 地址方式后，使用如下代码查看/etc/resolv.conf 文件的内容：

```
dranyu@dranyu-desktop:/etc/network$ cat /etc/resolv.conf
# Generated by NetworkManager
domain domain
search domain
nameserver 202.116.128.1
nameserver 202.116.128.2
```

其中 domain 行声明主机的域名为 domain；search 行指定域名的查询顺序；而 nameserver



行指定 DNS 服务器的 IP 地址，可有多行 nameserver，也就是说可指定多个 DNS 服务器。要访问指定 DNS 服务器来进行查询，需要设置此文件。

15.3.2 网络常用命令

Ubuntu 自身附带了网络管理的命令，如 ping、route、netstat 和 ifconfig 命令等。通过这些命令系统管理员可查看网络状况、配置网络连接的方式。了解网络管理的常用命令更有利与系统管理员理解图形操作的网络管理工具，提高网络管理的能力。它们之间的对应关系为：Ubuntu 中图形操作的网络管理工具对话框中“设备”选项卡上的功能调用了 ifconfig 命令；“Ping”选项卡上的功能调用了 ping 命令；“网络统计”标签上的功能调用了 netstat 命令；“端口扫描”选项卡上的功能调用了 route 命令。

1. ping 命令

ping 命令是用来测试网络的连通性的命令，而且是使用比较多的命令。无论是 Linux 还是 Windows 系列的操作系统都支持这一命令。ping 命令的执行原理为：先将 ICMP 中的“ECHO_REQUEST”数据包发送到远程网络的主机，然后接收到响应的信息，并通过响应信息来判断网络连接情况。

ping 命令的语法格式为：

```
ping [选项] IP 或域名
```

ping 命令常用的选项意义如下所示。

- -c<完成次数>：设置完成要求回应的次数。
- -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：只输出数值。
- -R：记录路由过程。
- -s<数据包大小>：设置数据包的大小。
- -v：详细显示指令的执行过程。

ping 命令的使用举例如下所示。

例 15-1：ping 命令测试与远程网络的连接状态

使用 ping 命令测试本机与 www.web3ds.com.cn 网络主机的连接状态，命令执行和结果显示如下：

```
dranyu@dranyu-desktop:~$ ping www.web3ds.com.cn
PING www.web3ds.com.cn (222.200.103.82) 56(84) bytes of data. //222.200.103.82 表示
www.web3ds.com.cn 的 IP 地址
64 bytes from 222.200.103.82: icmp_seq=1 ttl=64 time=0.959 ms
64 bytes from 222.200.103.82: icmp_seq=2 ttl=64 time=0.980 ms
64 bytes from 222.200.103.82: icmp_seq=3 ttl=64 time=0.794 ms
^C64 bytes from 222.200.103.82: icmp_seq=4 ttl=64 time=0.824 ms //^C 中断符
--- www.web3ds.com.cn ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 15050ms
rtt min/avg/max/mdev = 0.794/0.889/0.980/0.083 ms
```

显示结果中，222.200.103.82 表示远程主机 www.web3ds.com.cn 的 IP 地址；64bytes 表示 ICMP 数据包的大小；icmp_seq 表示数据包的序列号；“^C”符号表示中断 ping 的执行，此

时系统管理员按下“Ctrl+C”组合键；最后两行结果显示执行 ping 的统计信息，包括发送了 4 个数据包，接收了 4 个数据包，丢失了 0% 的数据包和所发时间为 15050ms 等信息。

例 15-2：限制发送 3 个数据包，数据包的大小为 32bytes

```
dranyu@dranyu-desktop:~$ ping -c 3 -s 32 192.168.1.104 //使用-c 和-s 选项
PING 192.168.1.104 (192.168.1.104) 32(60) bytes of data.
40 bytes from 192.168.1.104: icmp_seq=1 ttl=64 time=0.026 ms
40 bytes from 192.168.1.104: icmp_seq=2 ttl=64 time=0.036 ms
40 bytes from 192.168.1.104: icmp_seq=3 ttl=64 time=0.017 ms
--- 192.168.1.104 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms //数据传输状态统计信息
rtt min/avg/max/mdev = 0.017/0.026/0.036/0.008 ms
```

其中选项“-c 3”设置此次 ping 命令发送 3 个数据包，选项“-s 32”设置此次 ping 命令发送的数据包大小为 32bytes。ping 命令接收完成第三个数据包的响应信息后就结束命令。

补充说明

RTT (Round-Trip Time)：往返时延，在计算机网络中它也是一个重要的性能指标，它表示从发送端发送数据开始，到发送端收到来自接收端的确认（接收端收到数据后便立即发送确认），总共经历的时延。Min 表示最小的往返时延，Avg 表示平均往返时延，Max 表示最大的往返时延，Mdev 表示往返时延的算术平均偏差。

2. netstat 命令

netstat 命令用于显示网络连接、路由表和网络接口信息，显示网络状态信息。netstat 命令所显示的信息量很大，不使用参数难以阅读。netstat 命令的基本语法如下：

```
netstat [选项]
```

在不熟悉 netstat 命令的情况下，我们可使用--help 选项查看 netstat 命令的使用方法，如例 15-3 所示。

例 15-3：netstat 命令的帮助说明

```
dranyu@dranyu-desktop:~$ netstat --help
用法: netstat [-veenNcCF] [<Af>] -r          netstat {-V|--version|-h|--help}
        netstat [-vnNcaeol] [<Socket> ...]
        netstat { [-veenNac] -i | [-cnNe] -M | -s }
        -r, --route           显示路由表
        -i, --interfaces     显示接口信息表
        -g, --groups          显示多播组中的成员关系
        -s, --statistics      显示网络的统计信息
        -v, --verbose          显示详细信息
        --numeric-ports       忽略端口名称
        --numeric-users        忽略用户名
        -e, --extend            显示更多信息
...
<Socket>={-t|--tcp} {-u|--udp} {-w|--raw} {-x|--UNIX} --ax25 --ipx --netrom
<AF>=Use '-6|-4' or '-A <af>' or '--<af>'；默认: inet
列出所有支持的协议:
    inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
    x25 (CCITT X.25)
```

通过查看帮助信息，我们就可方便地使用 netstat 命令了。netstat 命令的常用例子如下所示。



例 15-4：使用 netstat 命令显示路由表信息

```
dranyu@dranyu-desktop:~$ netstat -r
内核 IP 路由表
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface
192.168.1.0     *              255.255.255.0 U        0 0          0 eth4
link-local      *              255.255.0.0   U        0 0          0 eth4
default         192.168.1.1   0.0.0.0     UG       0 0          0 eth4
```

其中各选项的含义如下所示。

- Destination：目的地网络或主机的 IP 地址，`default` 表示本机的默认网关地址。
- Gateway：此项表示传送数据包至目的地的网关或路由器，若值为“*”，则表示此项没有设置。
- Genmask：此项表示目的地网络的子网掩码，其值若为 255.255.255.255，则表示目的地为主机，而值为 0.0.0.0 则表示目的地为路由器。
- Flags：此项显示目前路由的状态的标志，其中包括了 U (UP) 表示此路由目前为启动状态；H (Host) 表示此网关为主机；G 表示此网关为路由器等。
- MSS：此项表示数据包经过路由器时，可传送的大小上限，以 Byte 为单位。
- Window：此项表示存放数据包的逻辑单位。
- irtt：irtt 是 initial round trip time 的缩写，指传送 TCP/IP 数据包的主机，等待由目的地传送 ACK 的时间。
- Iface：此项表示内乡适配卡的类型。

例 15-5：使用 netstat 命令显示网络接口信息

```
dranyu@dranyu-desktop:~$ netstat -i
Kernel Interface table
Iface MTU Met    RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth4  1500 0    51344    0    0 0      18510    0    0 0      BMRU
lo    16436 0   655466    0    0 0      655466    0    0 0      LRU
```

在结果列表中，Flg 项表示目前网卡的状况，其包括 B 表示设置为广播地址、L 表示设置为 Loopback 回环地址、R 表示目前执行中和 U 表示此网络接口已启用。而 MTU 项表示的是网络传输时数据包最大的传输单位。以 Rx- 为前缀的项表示网络接口接收数据包的情况，而以 Tx- 为前缀的项表示网络接口传输数据包时的情况，这些项的值的数字表示相应情况的数据包的数量。OK 表示正确的，ERR 表示错误的，DRP 表示遗失的和 OVR 表示超过流速限制的。

3. Ifconfig 命令

Ifconfig 命令是在网络配置使用比较多的一个网络管理工具。系统管理员可使用它显示和设置网卡的配置。Ifconfig 命令的使用语法为：

```
Ifconfig [选项] [设备名] [选项]
```

其常用的选项和设置参数如下。

- up：激活指定的接口，如 `Ifconfig up eth0` 表示激活 `eth0` 设备。
- down：关闭指定接口。该参数可以有效地阻止通过指定接口的 IP 信息流，如果想永久地关闭一个接口，我们还需要从核心路由表中将该接口的路由信息全部删除。
- netmask mask：此项表示为接口设置 IP 网络掩码。掩码可以是有前缀 0x 的 32 位十

六进制数，也可以是用点分开的 4 个十进制数。

- broadcast address：当使用了一个地址时，设置这个接口的广播地址。如果没有给出地址，就打开这个指定接口的 IFF_BROADCAST 选项。前面加上一个负号表示关闭这个选项。

以下将结合例子讲述 Ifconfig 的使用。

例 15-6：显示网卡信息

在不带参数的情况下，执行 Ifconfig 命令将显示当前已启动的网卡状态信息，代码如下所示：

```
dranyu@dranyu-desktop:~$ ifconfig
1 eth4      Link encap:以太网 硬件地址 00:0c:29:84:ce:5e
2          inet 地址:192.168.1.104 广播:192.168.1.255 掩码:255.255.255.0
3          inet6 地址: fe80::20c:29ff:fe84:ce5e/64 Scope:Link
4          UP BROADCAST RUNNING MULTICAST MTU:1500 跳点数:1
5          接收数据包:54943 错误:0 丢弃:0 过载:0 帧数:0
6          发送数据包:22006 错误:0 丢弃:0 过载:0 载波:0
7          碰撞:0 发送队列长度:1000
8          接收字节:10442130 (10.4 MB)  发送字节:2799048 (2.7 MB)
9          中断:18 基本地址:0x1400
10
11 lo       Link encap:本地环回
12          inet 地址:127.0.0.1 掩码:255.0.0.0
13          inet6 地址: ::1/128 Scope:Host
14          UP LOOPBACK RUNNING MTU:16436 跳点数:1
15          接收数据包:655466 错误:0 丢弃:0 过载:0 帧数:0
16          发送数据包:655466 错误:0 丢弃:0 过载:0 载波:0
17          碰撞:0 发送队列长度:0
18          接收字节:32881364 (32.8 MB)  发送字节:32881364 (32.8 MB)
```

结果显示了 eth4 和 lo 两个网络接口设备的信息。为了解说，我们在执行结果的前端添加了行号。第 1 行到第 9 行描述了 eth4 设备的信息。Link encapsulation 表示将信息划分为数据包传输分组的方法，如第 1 行显示了分组方法为以太网。第 2 行 inet 地址表示的是 IPv4 格式的主机 IP 地址；而第 3 行的 inet6 地址则表示是以 IPv6 为格式的主机 IP 地址。

从第 11 行到第 18 行描述了 lo 接口设备的信息。lo 接口设备是作为本地回环的接口。

补充说明

广播是指信息从本机上在同一时间传送到在同一网域中的所有计算机。广播地址也就是指定主机发送信息时使用的 IP 地址。广播地址是根据不同的网域而定的。如主机的 IP 地址是 A 类地址 20.34.68.30，那么它的广播地址为 20.255.255.255；若主机的 IP 地址为 B 类地址 128.34.26.68，那么它的广播地址为 128.34.255.255；相应地，若主机处于 C 类地址范围内，它的广播地址应为 xxx.xxx.xxx.255。

例 15-7：启动网卡

作为系统管理员掌握使用命令来启动网卡是有必要的。系统并不是没有错误的时候，如果系统不能自动启动网络设备，那么设备的启动工作就由系统管理员来完成。使用 Ifconfig 命令启动网卡功能的命令如下所示：

```
root@dranyu-desktop:/home/dranyu# ifconfig eth4 up
```

其中 eth4 表示网卡的网络接口；up 就是 Ifconfig 的选项，它表示启动 eth4 网卡的接口。



例 15-8：停止网卡

在修改网络配置文件的时候，系统管理员都需要先切断网络连接。使用 Ifconfig 命令停止网卡的命令如下所示：

```
root@dranyu-desktop:/home/dranyu# ifconfig eth4 down
```

其中 eth4 是网卡的网络接口；down 表示停止 eth4 网卡接口。

例 15-9：设置网卡

系统管理员可使用 Ifconfig 设置网卡的接口信息，设置网卡的静态 IP 地址。我们将 eth4 网卡的 IP 地址和子网掩码分别修改为 192.168.1.113 和 255.255.0.0，执行代码如下所示：

```
root@dranyu-desktop:/home/dranyu# ifconfig eth4
eth4      Link encap:以太网  硬件地址 00:0c:29:84:ce:5e
          inet 地址:192.168.1.104 广播:192.168.1.255 掩码:255.255.255.0
          inet6 地址: fe80::20c:29ff:fe84:ce5e/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 跳点数:1
            接收数据包:50 错误:0 丢弃:0 过载:0 帧数:0
            发送数据包:119 错误:0 丢弃:0 过载:0 载波:0
            碰撞:0 发送队列长度:1000
            接收字节:6539 (6.5 KB)  发送字节:23272 (23.2 KB)
            中断:18 基本地址:0x1400
root@dranyu-desktop:/home/dranyu# ifconfig eth4 192.168.1.113 netmask 255.255.0.0
root@dranyu-desktop:/home/dranyu# ifconfig eth4
eth4      Link encap:以太网  硬件地址 00:0c:29:84:ce:5e
          inet 地址:192.168.1.113 广播:192.255.255.255 掩码:255.255.0.0
          inet6 地址: fe80::20c:29ff:fe84:ce5e/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 跳点数:1
            接收数据包:20 错误:0 丢弃:0 过载:0 帧数:0
            发送数据包:64 错误:0 丢弃:0 过载:0 载波:0
            碰撞:0 发送队列长度:1000
            接收字节:2936 (2.9 KB)  发送字节:13837 (13.8 KB)
            中断:18 基本地址:0x1400
```

4. route 命令

route 命令主要功能是管理 Linux 系统内核中的路由表。它最大的用途就是用来设定静态的路由表项，通常是在系统用 Ifconfig 配置网络接口后，用它来设定主机或者一个网段的 IP 地址应该通过什么接口发送等。

route 命令有复杂的调用参数，其的基本语法格式如下所示：

```
route [选项]
```

其中主要常用参数说明如下所示。

- -v：使用冗余输出模式。
- -A family：指定特定的地址族（例如“inet”、“inet6”）。
- -n：使用数字显示的地址（例如 202.38.75.75）而不是去解释域名。
- -e：使用与 netstat 相同的输出格式。
- -ee：参数会产生很长的输出，包括内核路由表的几乎所有信息。
- -net：目标（target）是一个网段。
- -host：目标（target）是一个单独的主机。
- -F：显示内核 FIB 路由表。结果可能被-e 和-ee 参数改变。
- -C：显示内核中路由缓存信息。

- **del:** 删除一个路由表项。
- **add:** 增加一个路由表项。
- **target:** 配置的目的网段或者主机。可以是 IP，或者是网络或主机名。

下面举几个配置的例子。

例 15-10: 不带参数的 route 命令

```
root@dranyu-desktop:/home/dranyu# route
内核 IP 路由表
目标       网关       子网掩码      标志 跃点 引用 使用 接口
192.168.1.0   *          255.255.255.0    U     1     0      0 eth4
link-local     *          255.255.0.0     U     1000   0      0 eth4
default        192.168.1.1  0.0.0.0       UG    0     0      0 eth4
```

例 15-11: 添加路由表项

```
root@dranyu-desktop:/home/dranyu# route add -net 192.56.76.0 netmask 255.255.255.0
dev eth4
```

这一命令表示添加一条路由表项，网段 192.56.76.x 从接口设备 eth4 通过。

15.4 小结



本章概述了 Linux 的网络体系结构，简单介绍了 TCP/IP 协议族，分别介绍了 TCP 协议、UDP 协议和 IP 协议。向系统管理员讲述了图形化操作网络管理工具 GNOME-nettool 的用户，让系统管理员可方便查看本地的 IP 地址、了解当前网络的连接情况和查看主机的路由表信息。本章的最后讲述了网络配置的文件，以及如何使用命令来管理网络连接。

15.5 习题



1. 目前，常见的计算机网络有哪几种？
2. OSI 网络体系结构分为多少层？
3. TCP/IP 协议分为多少层？简述这些层的功能。
4. TCP 协议、UDP 协议和 IP 协议分别处于 TCP/IP 的哪一层？它们的功能是什么？
5. 使用网络工具查看本地的 IP 地址。
6. 编写/etc/network/interfaces 文件配置静态 IP 地址。
7. 将 192.168.1.112 的域名服务器添加到/etc/resolv.conf 文件。
8. 执行 ping www.google.com 命令查看网络状态信息，再执行 ping localhost 命令查看本地回环连接。
9. 使用 netstat 命令查看路由信息，然后再使用 route 命令查看路由信息，最后比较由这两个命令生成的结果。
10. 使用 Ifconfig 命令查看本地网络接口的状态信息。
11. 关闭网络连接后，使用 Ifconfig 命令设置网络的静态 IP 地址，然后启动网络连接。

Linux

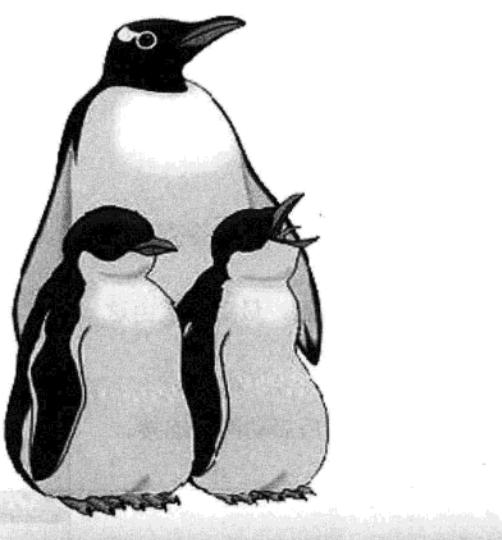
第 6 部分 服务器

第 16 章

网络应用服务器

Ubuntu 是多用户多任务的操作系统，同时它是一个具备强大的网络服务功能的网络操作系统，是商用网络服务器首选的操作系统之一。系统管理员可根据需要，在 Ubuntu 中搭建各种各样的网络服务器，如域名服务器、文件传输服务器、电子邮件服务器、数据库服务器、Web 服务器和代理服务器等。

本章主要介绍如何安装和配置 Bind9 域名服务器、vsFTPD 文件传输服务器和 Postfix 电子邮件服务器。





16.1 域名服务器 Bind9

BIND (Berkeley Internet Name Domain) 是 DNS 协议的一个实现，提供了 DNS 主要功能的开放实现，包括域名服务器，DNS 解析库函数，DNS 服务器运行调度所用的工具，是一款开放源码的 DNS 服务器软件，是由美国加州大学 Berkeley 分校开发和维护的。按照 ISC 的调查报告，BIND 是世界上使用最多最广泛的域名服务系统。

16.1.1 域名系统 DNS 简介

DNS 是 Domain Name System 的缩写，是域名系统，是一种主机名和 IP 地址的转换机制，它使用了一种分层的分布式数据库来处理 Internet 上的成千上万个主机名和 IP 地址的对应关系。

在域名系统中，每台计算机的域名由一系列用点分开的字母数字段组成。例如，网络上某台计算机的网址为 `www.ubuntu.org`，那么它的域名为 `ubuntu.org`。这些域名是有层次的，它最重要的部分位于右边。如`.org`的层次要比 `Ubuntu` 的层次高。此层次的基本单位就是域。域之间的关系是任何一个域最多属于一个上级域，但可以有多个或没有下级域；在同一个域下不能有相同的域名或主机名，但在不同的域中则可以有相同的域名或主机名。

1. DNS 的层次结构

在 Linux 文件系统中，根是以 “/” 开始的，而在 DNS 中，根是以 “.” 开始的，并被称为根域。DNS 的层次结构中，除了根域，还包括了顶级域、二级域以及相应的其他子域，如图 16-1 所示。

根域的下一级域名是顶级域，即在一个域名中最右边的那个词，如`.gov`、`.com`、`.cn`、`.org`、`.edu` 和`.net`等。目前，互联网上的域名体系中共有三类顶级域名：一是地理顶级域名，共有 243 个国家和地区的代码。例如`.CN` 代表中国，`.JP` 代表日本，`.UK` 代表英国等；另一类是类别顶级域名，共有 7 个：`.COM`（公司），`.NET`（网络机构），`.ORG`（组织机构），`.EDU`（美国教育），`.GOV`（美国政府部门），`.ARPA`（美国军方），`.INT`（国际组织）。由于互联网最初是在美国发展起来的，所以最初的域名体系也主要供美国使用，所以`.GOV`，`.EDU`，`.ARPA` 虽然都是顶级域名，但却是美国使用的。只有`.COM`，`.NET`，`.ORG`成了供全球使用的顶级域名。相对于地理顶级域名来说，这些顶级域名都是根据不同的类别来区分的，所以称之为类别顶级域名。随着互联网的不断发展，新的顶级域名也根据实际需要不断被扩充到现有的域名体系中来。新增加的顶级域名是`.BIZ`（商业），`.COOP`（合作公司），`.INFO`（信息行业），`.AERO`（航空业），`.PRO`（专业人士），`.MUSEUM`（博物馆行业），`.NAME`（个人）。

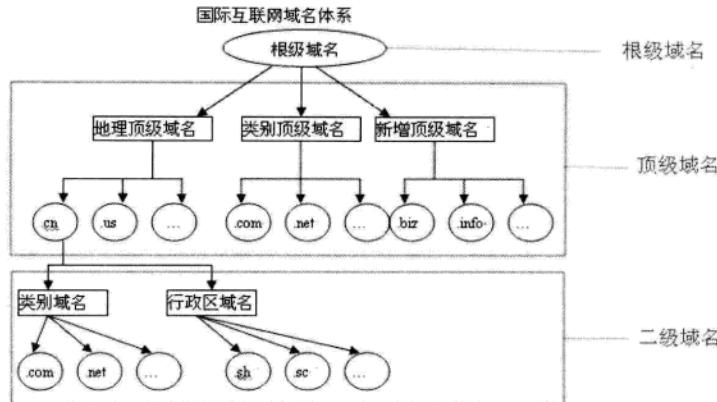


图 16-1 DNS 层次结构

而顶级域的下一级域是二级域，二级域是由国际域名组织为互联网中的个人或部门指定和登记的域。这个域的命名没有一个严格的规定，可由个人或申请此级域名的部门指定。

在二级域下又可进一步分为其他子域。最后子域中包括多台主机。

网络上出现的域名由上述的域所组成，如例 16-1 所示。

例 16-1：域名 bbs.ubuntu.org 的组成

此域名并没有显示根域。根域没有上级域，并且在 Internet 中都是默认的。因此，一般都不将根域显示出来。“.org”是一个顶级域，它表示申请此域名的组织是一个非营利性组织。接下来的是二级域“ubuntu”，其代表了此组织中的一台主机或域名服务器。而“bbs”表示此主机是在以“ubuntu.org”为服务器的一个网络里的一台主机。

2. DNS 解析过程

DNS 是典型的客户机/服务器模式结构，分为客户端和服务器端。客户端向服务器端发送解释域名的请求；服务器端查询层次分布式数据库，读取域名和与之对应的 IP 地址，最后将域名和 IP 地址的信息反馈给客户端。

3. 域名服务器的种类

在 Linux 下，按 DNS 服务的不同功能，可分为如下几种类型。

- **主服务器：**主服务器具有一个或几个区域的授权。区域就是一个或几个域的数据，也可以是一个域的部分数据。
- **辅服务器：**辅服务器也具有授权功能，作为主域名服务器的备份。它通过区域传送从主服务器获得所有的区域数据。
- **前向服务器：**前向服务器由其他的 DNS 服务器指定的一台处理其所有查询的 DNS 服务器。
- **从属服务器：**配置为使用前向服务器的 DNS 服务器，但它没有其他选择，必须使用前向服务器进行所有查询。注意这里不是指辅服务器。
- **缓存服务器：**通过自己的查询操作建立地址缓存的服务器。只用于缓存的服务器，没有自己的区域数据，只为客户端进行查询。

系统管理员使用 DNS 服务器配置软件，可将一台主机设置为不同类型的 DNS 服务器。

16.1.2 Bind 服务器简介

Bind 是 Berkeley Internet Name Domain Service 的简写，它是一款实现 DNS 服务器的开放源码软件。Bind 原本是美国 DARPA 资助伯克利大学（Berkeley）开设的一个研究生课题，后来经过多年的变化发展，已经成为世界上使用最为广泛的 DNS 服务器软件。目前，Internet 上绝大多数的 DNS 服务器都是用 Bind 来架设的。

Bind 经历了第 4 版、第 8 版和最新的第 9 版，第 9 版修正了以前版本的许多错误，并提升了执行时的效能。Bind 能够运行在当前大多数的操作系统平台之上。目前，Bind 软件由因特网软件联合会（Internet Software Consortium, ISC）这个非营利性机构负责开发和维护。ISC 的官方网站（<http://www.isc.org/>）包含了最新的错误修复和更新。

16.1.3 安装及启动 Bind9

Ubuntu 提供了 Bind9 的安装软件包，系统管理员通过新立得软件包管理器添加 Bind9 软件包，或者使用如下命令安装软件包：

```
dranyu@dranyu-desktop:~$ sudo apt-get install bind9
```

为了测试 DNS 服务器，我们还要添加一些管理工具，安装 bind9-host 和 bind9utils 软件包，执行如下命令：

```
dranyu@dranyu-desktop:~$ sudo apt-get install bind9-host bind9utils
```

安装 Bind9 软件包后，Bind9 服务自动开启。我们可使用/etc/init.d/bind9 命令来显示 bind9 服务的运行状态，命令如下所示：

```
dranyu@dranyu-desktop:~$ /etc/init.d/bind9 status
* bind9 is running
```

结果显示为 “*bind9 is running” 表示 Bind9 已经启动。

若 bind9 没有开启，我们可使用如下命令启动 Bind9 服务器：

```
dranyu@dranyu-desktop:~ $ sudo /etc/init.d/bind9 start
```

Bind9 服务停止和重启的命令如下所示：

```
$ sudo /etc/init.d/bind9 stop //停止 Bind9 服务器
$ sudo /etc/init.d/bind9 restart //重启 Bind9 服务器
```

16.1.4 配置 Bind9 服务器

Linux 上的域名服务器是由 named 守护进程来执行的，此进程会读取/etc/bind/named.conf 配置文件中的内容并且将主机名映射为 IP 地址的各种文件。安装 Bind9 域名服务器系统后，配置 bind9 服务器所需的文件都保存在 etc/bind 下，罗列这些配置文件的命令和结果显示如下所示：

```
dranyu@dranyu-desktop:/etc/bind$ ls -l
总用量 44
-rw-r--r-- 1 root root 237 2009-07-29 11:38 db.0
-rw-r--r-- 1 root root 271 2009-07-29 11:38 db.127
-rw-r--r-- 1 root root 237 2009-07-29 11:38 db.255
-rw-r--r-- 1 root root 353 2009-07-29 11:38 db.empty
-rw-r--r-- 1 root root 270 2009-07-29 11:38 db.local
```



```
-rw-r--r-- 1 root root 2878 2009-07-29 11:38 db.root
-rw-r--r-- 1 root bind 907 2009-07-29 11:38 named.conf
-rw-r--r-- 1 root bind 165 2009-07-29 11:38 named.conf.local
-rw-r--r-- 1 root bind 572 2009-07-29 11:38 named.conf.options
-rw-r----- 1 bind bind 77 2009-11-13 19:10 rndc.key
-rw-r--r-- 1 root root 1317 2009-07-29 11:38 zones.rfc1918
```

在结果中 `named.conf` 为主配置文件, `named.conf.local` 和 `named.conf.options` 这两个文件在其他的 UNIX/Linux 系统里默认是没有的, Ubuntu 则将各个部分都分开管理。而且还使用了以 `db` 为前缀的数据库文件分别保存资源记录。

1. 配置文件的编写格式

在配置 Bind9 服务器时, 我们需要理解的是主配置文件中指令编写的格式和资源记录的编写格式。

主配置文件`/etc/bind/named.conf` 中指令编写的格式如下所示:

```
Directive{
    parameter1;
    .....
    parameterN;
}
```

其中 `Directive` 是指用户所要使用的指令, 如 `options`、`zone` 等; `parameter1` 和 `parameterN` 是相应指令的参数, 而且在编写时要注意, 每一行只有一个参数, 每个参数后要以分号为结束。

安装 Bind9 之后, Bind9 已被配置为一个唯一高速缓存服务器, 它的主配置文件`/etc/bind/named.conf` 的内容如下所示:

```
// This is the primary configuration file for the BIND DNS server named.
//
.....
// If you are just adding zones, please do that in /etc/bind/named.conf.local
include "/etc/bind/named.conf.options"; //指定 options 指令的配置文件
// prime the server with knowledge of the root servers
zone "." { //注释: zone 是命令, “.” 表示域名的根域
    type hint; //type 是 zone 命令的一个参数, hint 表示 type 这个参数的值
    file "/etc/bind/db.root"; //file 是 zone 命令的一个参数, “/etc/bind/db.root”表示 file
参数的值
};
// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912
zone "localhost" { //注释: localhost 表示本地域, 此节正向解析 localhost 域
    type master;
    file "/etc/bind/db.local";
};
zone "127.in-addr.arpa" { //此节反向解释 IP 地址为 127.*.*.* 的主机, *是匹配符
    type master;
    file "/etc/bind/db.127";
};
.....
include "/etc/bind/named.conf.local"; //指定系统管理员配置的本地 DNS 的 zone 指令
```

其中 `type` 项的值中, `master` 表示定义的是主域名服务器; `slave` 表示定义的是辅助域名服务器; `hint` 表示是互联网中根域名服务器。在配置主 DNS 服务器时, 我们修改`/etc/bind/named.conf.local` 而不修改`/etc/bind/named.conf`。

2. 资源记录的编写

一个域的 DNS 数据库是一个文本文件集合，并且由该域的主域名服务器的系统管理员维护。这些文本文件成为区域文件，也就是由用户在“named.conf”或“named.conf.local”中定义的 file 属性值。它们包括 2 种类型的项：分析器命令（如\$ORIGIN 和\$TTL）和资源记录（Resource Records）。其中资源记录是数据库的真实部分，而分析器只是提供输入记录的一些简便途径。

资源记录的编写格式为：

```
[name] [ttl] [class] type data
```

其中各选项的含义如下所示。

- name 字段：表示一个主机或者一个域，通常来说是“@”，其代表本域。
- ttl 字段：以秒为单位表示存活时间。
- class 字段：一般使用“IN”值，对应的是“Internet”。
- type 字段的值：可设置为 SOA、A、NS、PTR 和 CHANE 等值。
- data 字段的值：会根据 type 字段值的不同而不同。

在 type 字段值的含义如下所示。

- SOA 记录：指示该区的权威。
- NS 记录：列出该区的一个名字服务器。
- A 记录：名字到地址的映射（也就是正向解析）。
- PTR 记录：地址到名字的映射（也就是反向解析）。
- CNAME：规范名字（别名记录）。

以编写 A 资源记录为例，如例 16-2 所示。

例 16-2：设置 A 资源记录

```
@ IN A 192.168.1.102 //设置地址记录
```

@表示本域，IN 是表示连接的网络为 Internet，A 表示此资源记录为地址资源记录，而 192.168.1.102 则为对@的解释数据。

在 Ubuntu 系统中，这些区域文件都是以 db 为前缀，如 127 开始的 IP 的反解析区域文件为 db.127，localhost 域名的正向解析区域文件为 db.local。db.127 文件里的内容如下所示：

```
;
; BIND reverse data file for local loopback interface
;

$TTL 604800
@ IN SOA localhost. root.localhost. (      //定义 SOA 资源记录
          1           ; Serial
          604800       ; Refresh
          86400        ; Retry
          2419200      ; Expire
          604800 )     ; Negative Cache TTL
;

@ IN NS localhost.      //定义 NS 资源记录
1.0.0 IN PTR localhost. //定义 PTR 资源记录
```

其中“;”标识文件内的解释行。SOA 设置是被括在“()”之间的 5 组数字，主要作为和服务器同步 DNS 资料所使用的资料，这组数字的字段名分别为：Serial、Refresh、Retry、



Expire 和 Negative Cache TTL。

- **Serial:** 其格式通常会是“年月日+修改次序”。当 slave 要进行资料同步的时候，会比较这个号码。如果发现在这里的号码比它那边的数值“大”，就进行更新，否则忽略。不过设 Serial 有一个地方您要留意：不能超过 10 位数字！
- **Refresh:** 这里是告诉 slave 要隔多久进行资料同步。是否同步则需要看 Serial 的比较结果。
- **Retry:** 如果 slave 在进行更新失败后，要隔多久再进行重试。
- **Expire:** 这是记录逾期时间：当 slave 一直未能成功与 master 取得联系，那到这里就放弃 retry，同时这里的资料也将标识为过期(expired)。
- **Minimum:** 这是最小默认 TTL 值，如果您在前面没有用“\$TTL”来定义，就会以此值为准。

注意

Localhost 是域名，在编写资源记录时，域名要以“.”结尾，如 SOA 资源记录中的“localhost.”。SOA 资源记录中的“root.localhost.”表示系统管理员的邮箱地址，它与“root@localhost”拥有同样的含义。以下的 SOA 资源记录中的系统管理员的邮箱地址写法与此处相同。一定要记住将“@”转化为“.”，而且必须以“.”结尾。

3. 配置主域名服务器

假设一台主机的 IP 地址为 192.168.1.102，我们要将其配置为 web3ds.org 域的主域名服务器。配置过程如下所示：

(1) 配置启动文件/etc/bind/named.conf.local。在此配置文件中，添加如下代码：

```
zone "web3ds.org" IN {           // 使用 zone 命令定义一个区域 "web3ds.org"，解析 web3ds.org  
域的主机  
    type master;                // 定义此服务器为主 DNS 服务器  
    file "/etc/bind/db.web3ds.org"; // 指定正向解析数据库文件的路径  
};  
zone "1.168.192.in-addr.arpa" {    // 定义反向区域  
    type master;                // 定义此服务器为主 DNS 服务器  
    file "/etc/bind/db.192.168.1"; // 指定反向解析数据库文件的路径  
};
```

(2) 配置正向解析数据库文件 db.web3ds.org。使用 vi、vim 或 Gedit 等编辑工具添加配置文件 db.web3ds.org，并保存到/etc/bind/目录下。此文件的内容如下所示：

```
$TTL 86400          // 设定 TTL—存活时间  
@ IN SOA dranyu.web3ds.org. root.dranyu.web3ds.org. (      // 定义 SOA 资源记录  
    1           ; Serial  
    604800      ; Refresh  
    86400       ; Retry  
    2419200     ; Expire  
    86400 )     ; Negative Cache TTL  
;  
@ IN NS dranyu.web3ds.org.          // 定义 NS 资源记录  
dranyu IN A 192.168.1.102          // 设置地址记录  
www IN CNAME dranyu.web3ds.org.    // 设置别名记录
```

(3) 配置反向解析数据库文件 db.192.168.1。在目录/etc/bind 下添加文件 db.192.168.1，

此文件的内容如下所示：

```
$TTL 604800 //设定 TTL-存活时间
@ IN SOA dranyu.web3ds.org. root.dranyu.web3ds.org. ( //定义 SOA 资源记录
    200      ; Serial
    604800    ; Refresh
    86400     ; Retry
    2419200   ; Expire
    604800 )   ; Negative Cache TTL
;
@ IN NS dranyu.web3ds.org. //定义 NS 资源记录
102 IN PTR dranyu.web3ds.org. //设置反向地址指针记录
```

(4) 重启 Bind9 服务。完成配置后，需要将 Bind9 服务器重启才能启动设置。重启 Bind9 服务器的命令和执行结果如下所示：

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/bind9 restart
* Stopping domain name service... bind9 [ OK ]
* Starting domain name service... bind9 [ OK ]
```

补充说明

在 named.conf.local 中定义“web3ds.org”区域和“1.168.192.in-addr.arpa”区域时，我们要理解这两个都是一个域，并不代表一台主机。如 1.168.192.in-addr.arpa 可解析 IP 地址在 192.168.1.0~192.168.1.255 之间。“zone ‘web3ds’.org”表示我们目前的网域在“web3ds.org”下。在配置解析数据库文件时，我们使用 dranyu.web3ds.org 和 192.168.1.102 表示提供 DNS 服务的主机。

4. 配置辅域名服务器

配置辅域名服务器要比配置主域名服务器来得简单。在配置辅域名服务器时，只需对配置文件/etc/bind/named.conf.local 进行配置，并不需要配置数据库文件，数据库文件将从主域名服务器中自动获得。配置辅域名服务器的具体操作步骤如下所示：

(1) 使用 vi 或 Gedit 等文本编辑器，打开/etc/bind/named.conf.local 文件，并编辑此文件，添加如下代码：

```
zone "web3ds.org" IN { //定义区域 “web3ds.org”
    type slave; //定义此服务器为辅 DNS 服务器
    file "/etc/bind/db.web3ds.org"; //指定正向解析数据库文件的路径
    masters {192.168.1.102}; //指定主域名服务器的 IP 地址
};
zone "1.168.192.in-addr.arpa" { //定义反向区域
    type slave; //定义此服务器为辅 DNS 服务器
    file "/etc/bind/db.192.168.1"; //指定反向解析数据库文件的路径
    masters {192.168.1.102}; //指定主域名服务器的 IP 地址
};
```

(2) 重启 Bind9 服务。完成配置后，需要将 Bind9 服务器重启才能启动设置。重启 Bind9 服务器的命令和执行结果如下所示：

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/bind9 restart
* Stopping domain name service... bind9 [ OK ]
* Starting domain name service... bind9 [ OK ]
```



补充说明

一个域的主域名服务器和辅域名服务器不可同时部署在同一台主机上。因此，在配置辅域名服务器使用的主机不是配置主域名服务器的主机，在上述中辅域名服务器并不使用 192.168.1.102 的 IP 地址。

16.1.5 测试 Bind9 服务器

安装或配置域名服务器后，我们需要测试安装或重新配置的服务器是否可运行。这时我们就需要使用到一些测试 DNS 服务器的测试工具，这些常用的测试工具包括 host 命令和 nslookup 命令。这些测试工具分别包含在 bind9-host 和 bind9utils 软件包中，在安装 Bind9 域名服务器系统时，我们已安装了这些软件包。本节介绍使用 host 和 nslookup 命令测试 DNS 域名服务器的配置。

在测试域名服务器前，需要先修改客户端的一个配置文件 /etc/resolv.conf，使客户端应用配置好的 DNS 服务器，以主域名服务器为例。使用 Gedit 文本编辑器打开 /etc/resolv.conf，将其内容修改为如下所示的代码：

```
# Generated by NetworkManager
nameserver 192.168.1.102
```

第一行是注释行，表示此文件的内容是由 NetworkManager 程序自动生成的；第二行指定客户端使用的 DNS 服务器的 IP，此处指定为“192.168.1.102”。

补充说明

在测试前一定要设置客户端使用的 DNS 服务器，否则不能解析 Bind9 中配置的域名信息。配置 /etc/resolv.conf 文件，主要就是告诉系统要将域名解析的数据包发送何处进行解析。将 nameserver 设置为 192.168.1.102，即将在 192.168.1.* 区域的域名解析权赋予给 192.168.1.102 主机。

1. 使用 host 命令测试 DNS

host 命令能把一个主机名解析到一个网际地址或把一个网际地址解析到一个主机名。它的基本语法如下所示：

```
host [选项] 参数
```

常用选项如下。

- **-a:** 此项显示主机的所有信息。
- **-c Class:** 当搜索非网际数据时指定要查找的类。Class 的值就是类名，有效类为：IN 网络类、CHAOS 、HESIOD、ANY 和* 。
- **-d:** 此项表示打开调试方式。
- **-t Type :** 指定要查询的记录类型。有效类型为：A 表示主机的网际地址；CNAME 表示别名的规范名称；MX 表示邮件交换器；NS 表示指定范围的名称服务器；PTR 表示如果查询的是一个网际地址则为主机名；否则为其他信息的指针；SIG 表示签名记录，SOA 表示域的“授权开始”信息。
- **-v:** 显示主机的详细信息。

- -w：永远等待 DNS 服务器的一个回答。
- -z：使用显示资源记录信息的新输出。

举例如下。

例 16-3：反向查询域名

```
root@dranyu-desktop:/home/dranyu# host 192.168.1.102
102.1.168.192.in-addr.arpa domain name pointer dranyu.web3ds.org.
//192.168.1.102 是 IP 地址, dranyu.web3ds.org 就是此 IP 对应的域名
```

例 16-4：正向查询主机地址

```
root@dranyu-desktop:/home/dranyu# host dranyu.web3ds.org
dranyu.web3ds.org has address 192.168.1.102
//与域名 dranyu.web3ds.org 相对应的 IP 地址是 192.168.1.102
```

例 16-5：查询 NS 资源记录信息

```
root@dranyu-desktop:/home/dranyu# host -t NS web3ds.org // -t NS 指定查询 NS 类型的
资源记录
web3ds.org name server dranyu.web3ds.org.
// -t 选项指定查询资源记录的类型
```

例 16-6：查询 SOA 资源记录信息

```
root@dranyu-desktop:/home/dranyu# host -t SOA web3ds.org // -t SOA 指定查询 SOA 类型的
资源记录
web3ds.org has SOA record dranyu.web3ds.org. root.dranyu.web3ds.org. 1 604800 86400
2419200 86400
// -t 选项指定查询资源记录的类型, 显示的结果与 /etc/bind/db.web3ds.org 数据库文件的内容相同
```

例 16-7：列出主机别名的信息

```
root@dranyu-desktop:/home/dranyu# host www.web3ds.org
www.web3ds.org is an alias for dranyu.web3ds.org.
dranyu.web3ds.org has address 192.168.1.102
//www.web3ds.org 是 dranyu.web3ds.org 域名的别名, 而它的 IP 是 192.168.1.102
```

例 16-8：列出与指定主机名相关的资源记录的详细信息

```
root@dranyu-desktop:/home/dranyu# host -a dranyu.web3ds.org
Trying "dranyu.web3ds.org"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24231
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;dranyu.web3ds.org.           IN  ANY   //发送信息出去

;; ANSWER SECTION:
dranyu.web3ds.org.      86400    IN  A    192.168.1.102 //A 资源记录

;; AUTHORITY SECTION:
web3ds.org.          86400    IN  NS   dranyu.web3ds.org. //返回认证信息
```

Received 65 bytes from 192.168.1.102#53 in 1 ms //接收到返回信息

2. 使用 nslookup 命令测试 DNS

nslookup 命令是一个监测网络中 DNS 服务器是否能正确实现域名解析的命令行工具。nslookup 可使用命令模式或命令行提示符模式运行。命令模式的基本语法为：

nslookup [选项] 域名或 IP 地址

可使用 man nslookup 命令来查看 nslookup 命令模式的详细使用说明。此处主要讲述



nslookup 命令的命令提示符模式，应用举例如下所示。

例 16-9：启动 nslookup 命令，并反向解释 IP 地址，查询域名

```
root@dranyu-desktop:/home/dranyu# nslookup //输入 nslookup 进入 nslookup 的字符交互界面
> 192.168.1.102 //输入 IP 地址，如 192.168.1.102，然后按回车键
Server: 192.168.1.102 //域名服务器的主机名
Address: 192.168.1.102#53 //域名服务器的 IP 地址，与所使用的端口
```

```
102.1.168.192.in-addr.arpaname = dranyu.web3ds.org. //通过反向解释，将 IP 解释为域名
```

例 16-10：正向查询主机地址

```
> dranyu.web3ds.org //输入主机的域名，如 dranyu.web3ds.org，查询主机 IP
Server: 192.168.1.102
Address: 192.168.1.102#53
```

```
Name: dranyu.web3ds.org
Address: 192.168.1.102
```

例 16-11：显示当前设置的所有数值参数

```
> set all //设置 nslookup 命令的运行环境
Default server: 192.168.1.102
Address: 192.168.1.102#53

Set options:
  novc          nodebug        nod2
  search        recurse
  timeout = 0    retry = 3    port = 53
  querytype = A      class = IN
  srchlist =
//其中没有等号的属性名表示没有设置值，而有值的就是没有设置
```

例 16-12：查询 NS 资源记录

```
> set type=NS //设置查询的资源记录类型为 NS
> web3ds.org
Server: 192.168.1.102
Address: 192.168.1.102#53
web3ds.org nameserver = dranyu.web3ds.org.
```

例 16-13：查询 SOA 资源记录

```
> set type=SOA //设置查询的资源记录类型为 SOA
> web3ds.org
Server: 192.168.1.102
Address: 192.168.1.102#53

web3ds.org
  origin = dranyu.web3ds.org
  mail addr = root.dranyu.web3ds.org
  serial = 1
  refresh = 604800
  retry = 86400
  expire = 2419200
  minimum = 86400
```

例 16-14：查询别名信息

```
> set type=CNAME //设置查询资源记录的类型为 CNAME
> www.web3ds.org
Server: 192.168.1.102
Address: 192.168.1.102#53
```

```
www.web3ds.org canonical name = dranyu.web3ds.org.
```

例 16-15：查询 web3ds.org 域的所有配置

```
> set type=any //设置所有类型的资源记录
> web3ds.org
Server: 192.168.1.102
Address: 192.168.1.102#53

web3ds.org
origin = dranyu.web3ds.org
mail addr = root.dranyu.web3ds.org
serial = 1
refresh = 604800
retry = 86400
expire = 2419200
minimum = 86400
web3ds.org nameserver = dranyu.web3ds.org.
```

例 16-16：查询反向域的所有配置

```
> 1.168.192.in-addr.arpa //输入反向域请求，按下回车键
Server: 192.168.1.102
Address: 192.168.1.102#53

1.168.192.in-addr.arpa
origin = dranyu.web3ds.org
mail addr = root.dranyu.web3ds.org
serial = 200
refresh = 604800
retry = 86400
expire = 2419200
minimum = 604800
1.168.192.in-addr.arpa nameserver = dranyu.web3ds.org.
>exit //退出 nslookup
```



16.2

文件传输服务器 vsFTPD

Linux 下的 FTP 软件很多，常见的有 vsFTP、ProFTP、Wu-FTP 等，而 vsFTP 在安全性、高性能及稳定性 3 个方面表现比较出色。它提供的主要功能包括虚拟 IP 设置、虚拟用户、Standalone、inetd 操作模式、强大的单用户设置能力及带宽限流等。在安全方面，它从原理上修补了大多数 ProFTP、Wu-FTP 的安装缺陷，并使用了安全编码技术解决了缓冲问题，并能有效地避免“globbing”类型的拒绝服务攻击。

16.2.1 FTP 服务器的特性

FTP 是 TCP/IP 协议组中的协议之一，是英文 File Transfer Protocol 的缩写。该协议是 Internet 文件传送的基础，它由一系列规格说明文档组成，目标是提高文件的共享性，提供非直接使用远程计算机，使存储介质对用户透明和可靠高效地传送数据。简单地说，FTP 就是完成两台计算机之间的复制，从远程计算机复制文件至自己的计算机上，称之为“下载（download）”文件。若将文件从自己计算机中复制到远程计算机上，则称之为“上载（upload）”



文件。在 TCP/IP 协议中，FTP 标准命令 TCP 端口号为 21，Port 方式数据端口为 20。FTP 协议的任务是从一台计算机将文件传送到另一台计算机，它与这两台计算机所处的位置、连接的方式甚至是否使用相同的操作系统无关。假设两台计算机通过 FTP 协议对话，并且能访问 Internet，你可以用 FTP 命令来传输文件。每种操作系统使用上有一些细微差别，但是每种协议基本的命令结构是相同的。

简单地说，支持 FTP 协议的服务器就是 FTP 服务器。虽然目前 FTP 服务器已经被网页服务器取代了部分功能，但它仍然有自己的特性：

(1) FTP 服务器允许用户使用交互的方式进行访问，用户登录后不但能够进行文件的传输，并且还能够在权限允许的情况下进行文件的建立和删除。

(2) FTP 服务器传输信息非常稳定，它主要使用 20 端口和 19 端口，其中 20 端口主要服务传输数据，而 19 端口主要用来控制传输过程，这样使 FTP 区别于其他的通信协议。因此，其具有很好的稳定性。

(3) FTP 服务器可以指定登录的用户，并且设置其权限，保证了服务器的安全性。

16.2.2 FTP 用户类型

在考虑 FTP 服务器安全性工作的时候，首先要考虑的就是谁可以访问 FTP 服务器。所以在 vsFTPD 服务器软件中，默认提供了三类用户。不同的用户对应着不同的权限与操作方式。

第一类是本地用户。这类用户是指在 FTP 服务上拥有账号。当这类用户登录 FTP 服务器的时候，它默认的主目录就是其账号命名的目录。但是，它还可以变更到其他目录中去。如系统的主目录等。

第二类账户是虚拟用户。在 FTP 服务器中，我们往往会给不同的部门或者某个特定的用户设置一个账户。但是，这个账户有个特点，就是其只能访问自己的主目录。服务器通过这种方式来保障 FTP 服务上其他文件的安全性。这类账户，在 vsFTPD 软件中就叫做虚拟用户。拥有这类用户的账户，只能访问其主目录下的目录，而不得访问主目录以外的文件。

第三类账户是匿名用户，这也是我们通常所说的匿名访问。这类用户是指在 FTP 服务器中没有指定账户，但是他仍然可以进行匿名访问某些公开的资源。

在组建 FTP 服务器的时候，我们就需要根据用户的类型，对用户进行归类。默认情况下，vsFTPD 服务器会把建立的所有账户都归属为本地用户。但是，这往往不符合企业安全的需要。因为这类用户不仅可以访问自己的主目录，而且还可以访问其他用户的目录。这就给其他用户所在的空间带来一定的安全隐患。所以，企业要根据实际情况，修改用户所在的类别。

16.2.3 安装 vsFTPD 服务器，以及它的基本操作

由于默认的情况下，Ubuntu 并不会安装 vsFTPD 服务器软件包 vsFTPD。所以，在使用这一 FTP 服务器的时候需要安装它。安装 vsFTPD 服务器软件包的方法有多种，可使用新立得软件包管理器工具安装、使用 apt-get 命令在命令行中安装或者使用 make 命令编译 vsFTPD 源码包。

1. 安装 vsFTPD 服务器

使用新立得软件包管理器安装 vsFTPD 服务器的方法为：打开新立得软件包管理器，查

找“vsftpd”软件包，然后安装此软件包。

若使用命令行终端安装 vsFTPD 服务器，则在提示符后输入如下命令：

```
root@dranyu-desktop:/home/dranyu $sudo apt-get install vsftpd
```

2. 启动 vsFTPD 服务器

安装 vsFTPD 服务器后，服务器会自动运行，也就是说其处于运行状态。系统管理员同样也可使用命令行来启动 vsFTPD 服务器，只要在终端的命令提示符后输入如下代码即可：

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/vsftpd start
*                                         Starting          FTP           server:      vsftpd
/usr/sbin/vsftpd already running.                                     [fail]
```

出现以上结果，是因为已有一个 vsFTPD 在系统中运行。

3. 关闭 vsFTPD 服务器

如果系统管理员需要关闭 vsFTPD 服务器，可在终端的命令提示符输入如下代码：

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/vsftpd stop
* Stopping FTP server: vsftpd                                     [ OK ] //表示 vsftpd 已停止
```

4. 重启 vsFTPD 服务器

若 vsFTPD 服务器已停止，执行重启命令和显示结果如下所示：

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/vsftpd restart
* Stopping FTP server: vsftpd                                     No
/usr/sbin/vsftpd found running; none killed.
* Starting FTP server: vsftpd                                     [ OK ]
* Starting FTP server: vsftpd                                     [ OK ]
```

若 vsFTPD 服务器处于运行状态，执行重启命令和显示结果如下所示：

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/vsftpd restart
* Stopping FTP server: vsftpd                                     [ OK ]
* Starting FTP server: vsftpd                                     [ OK ]
```

16.2.4 配置 vsFTPD 服务器

安装 vsFTPD 软件包后，FTP 服务器也就可使用了，只是此时的 FTP 服务器只允许匿名用户以只读的方式访问。在文件管理器中的地址栏上输入 `ftp://127.0.0.1` 就可访问 `/home/ftp` 中的内容。

注意：

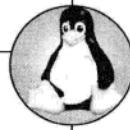
在安装 vsFTPD 的时候，`/home/ftp` 被自动创建。此目录是默认的 ftp 根目录，用户输入 `ftp://127.0.0.1` 时的指向目录。

1. 配置文件

在 Ubuntu 中 vsFTPD 的默认配置文件有 2 个，包括 `/etc/vsftpd.conf` 和 `/etc/ftpusers`。其中 `/etc/vsftpd.conf` 是 vsFTPD 的主配置文件；`/etc/ftpusers` 文件中指定了哪些用户不能访问 FTP 服务器。

`/etc/vsftpd.conf` 的默认配置如下所示：

```
listen=YES          //开启 FTP 服务器功能，使 vsftpd 处于独立启动模式
anonymous_enable=YES //允许匿名登录
#local_enable=YES   //此句话不起作用，去掉#后表示允许本地用户登录
#writable_enable=YES //设置本地用户的写权限
#local_umask=022    //设置本地用户的文件生成掩码为 022
#anon_upload_enable=YES //设置匿名用户上传权限，YES 允许，NO 拒绝
```



Ubuntu Linux 从初学到精通

```
#anon_mkdir_write_enable=YES //设置匿名用户创建新目录权限, YES 允许, NO 拒绝  
dirmessage_enable=YES //为 YES 则进入目录时显示此目录下由 message_file 选项指定的文本文件(默认  
为.message)的内容  
xferlog_enable=YES //激活上传和下载的日志  
connect_from_port_20=YES //启用 FTP 数据端口 20 的数据连接  
#xferlog_file=/var/log/vsftpd.log // 使用上传 / 下载 日志, 日志文件默认为  
/var/log/vsftpd.log, 可以通过 xferlog_file 选项修改  
#xferlog_std_format=YES //设置日志使用标准 xferlog 格式  
#ftpd_banner=Welcome to blah FTP service. //设置欢迎信息  
#chroot_local_user=YES //将本地用户锁定在自家目录中, YES 锁定, NO 不锁定  
#chroot_list_enable=YES //设置是否启动 chroot 列表, 在 chroot 列表中的用户将不被锁定  
#chroot_list_file=/etc/vsftpd.chroot_list //设置 chroot 列表文件
```

在上述代码中, 此处为了说明忽略了一部分注释, 以“#”开头的行为注释行, 而“//”后面的字符串是对代码的解说。

/etc/ftpusers 文件的默认内容如下所示:

```
# /etc/ftpusers: list of users disallowed FTP access. See ftpusers(5).  
root  
daemon  
bin  
.....  
news  
uucp  
nobody
```

上述用户是系统中的系统用户, 不被允许登录 ftp 服务器。

Ubuntu 下的 vsFTPD 应用程序, 为简化系统管理员的配置操作将 FTP 服务器的大部分配置选项都集中在/etc/vsftpd.conf 配置文件中, 所以在使用 vsFTPD 服务器时, 系统管理员只需修改/etc/vsftpd.conf 配置文件就可实现 FTP 的大部分功能。

2. 禁止匿名访问和允许本地用户访问且修改内容

假设安装 vsFTPD 应用程序的主机 IP 地址为 192.168.1.102, 现在需要禁止匿名用户的使用, 开启本地用户访问 FTP, 并且可上传和修改内容, 具体的操作步骤如下所示:

(1) 使用文本编辑器 Gedit 编辑主配置文件/etc/vsftpd.conf, 在默认的配置内容中, 将相应设置项的值修改为如下代码:

```
anonymous_enable=NO //匿名用户不可登录  
local_enable=YES //允许本地用户访问 FTP 服务器  
write_enable=YES //允许登录用户写磁盘  
local_umask=022 //设置本地用户的文件生成掩码为 022
```

(2) 重启 vsFTPD 服务器。重启 FTP 服务器的命令如下所示:

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/vsftpd restart  
* Stopping FTP server: vsftpd [ OK ]  
* Starting FTP server: vsftpd [ OK ]
```

(3) 测试以 dranyu 的身份访问 FTP。执行命令和结果显示如下所示:

```
root@dranyu-desktop:/home# ftp 192.168.1.102 //连接 192.168.1.102 ftp 服务器  
Connected to 192.168.1.102.  
220 (vsFTPD 2.0.7)  
Name (192.168.1.102:dranyu): dranyu //输入本地用户名 dranyu  
331 Please specify the password.  
Password: //输入本地用户 dranyu 的密码  
230 Login successful.
```

```

Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls                                //列举 192.168.1.102 中用户 dranyu 主文件夹中的文件和目录
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x  3 1000   1000      4096 Nov 15 01:58 Documents
drwxr-xr-x  2 1000   1000      4096 May 30 16:29 Pictures
....
drwxr-xr-x  3 1000   1000      4096 Jun 27 12:33 software
ftp> cd Documents           //进入 192.168.1.102 下/home/dranyu/Documents 目录
250 Directory successfully changed.
ftp> delete quota.txt //删除 192.168.1.102 中的/home/dranyu/Documents/quota.txt 文件
250 Delete operation successful. //成功删除文件

```

(4) 测试以匿名用户的身份访问 FTP。执行命令和结果显示如下所示：

```

root@dranyu-desktop:/home# ftp 192.168.1.102          //连接 192.168.1.102 ftp 服务器
Connected to 192.168.1.102.
220 (vsFTPd 2.0.7)
Name (192.168.1.102:dranyu):                      //需要使用匿名用户，所以不输入用户名
331 Please specify the password.
Password:                                            //不输入密码，直接按下回车键
530 Login incorrect.                                //登录失败
Login failed.
ftp> ls                                         //使用列举文件命令 ls
530 Please login with USER and PASS.               //列举文件失败，需要用户名和密码
ftp: bind: Address already in use

```

16.2.5 配置本地组访问的 FTP

利用 Ubuntu 系统中用户权限的机制，设置用户组和用户之间的关系。这样可实现同在用户组 project1 中的 soft1、soft2 和 soft3 三个用户拥有不同的访问权限，即 soft1 拥有对 FTP 的读写权限，而 soft2 和 soft3 对 FTP 拥有只读权限，具体操作步骤如下所示：

(1) 创建 project 组，以及组的目录，执行的命令如下所示：

```

root@dranyu-desktop:/home/dranyu# groupadd project
root@dranyu-desktop:/home/dranyu# mkdir /usr/project

```

(2) 创建 3 个用户，并为 3 个用户添加密码，命令如下所示：

```

root@dranyu-desktop:/usr# useradd -G project -d /usr/project soft1
root@dranyu-desktop:/usr# useradd -G project -d /usr/project soft2
root@dranyu-desktop:/usr# useradd -G project -d /usr/project soft3
root@dranyu-desktop:/usr/project# passwd soft1
输入新的 UNIX 口令:          //输入密码
重新输入新的 UNIX 口令:      //输入确认密码
passwd: 已成功更新密码
root@dranyu-desktop:/usr/project# passwd soft2
输入新的 UNIX 口令:          //输入密码
重新输入新的 UNIX 口令:      //输入确认密码
passwd: 已成功更新密码
root@dranyu-desktop:/usr/project# passwd soft3
输入新的 UNIX 口令:          //输入密码
重新输入新的 UNIX 口令:      //输入确认密码
passwd: 已成功更新密码

```

其中-G project 指定用户的所属的组，-d /usr/project 指定用户的主文件夹。



Ubuntu Linux 从初学到精通

(3) 修改/usr/project 目录的权限，执行命令如下所示：

```
root@dranyu-desktop:/usr/project# chown soft1.project /usr/project  
root@dranyu-desktop:/usr/project# chmod 750 /usr/project
```

第一行命令设置/usr/project 目录所属组为 project，所属用户为 soft1。第二行命令设置 /usr/project 的所属用户 soft1 拥有控制/usr/project 的所有权限；组 project 里的其他成员对 /usr/project 不具有写权限；组 project 以外的用户对/usr/project 没有任何权限。

(4) 配置/etc/vsftpd.conf 文件。设置指定本地用户访问自己的主文件夹，则需要设置以下几个选项：

```
local_enable=YES  
write_enable=YES  
chroot_local_user=YES
```

(5) 重启 vsFTPD 服务器。重启 FTP 服务器的命令如下所示：

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/vsftpd restart  
* Stopping FTP server: vsftpd [ OK ]  
* Starting FTP server: vsftpd [ OK ]
```

(6) 测试 soft1、soft2 和 soft3 的读写权限，执行命令和结果显示如下所示：

```
root@dranyu-desktop:~# ftp 192.168.1.102 //连接 192.168.1.102ftp 服务器  
Connected to 192.168.1.102.  
220 (vsFTPd 2.0.7)  
Name (192.168.1.102:dranyu): soft1 //以 soft1 用户名登录  
331 Please specify the password.  
Password: //输入 soft1 用户的密码  
230 Login successful. //成功登录  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> put file1.txt //上传一个保存在 root 主文件夹的 file1.txt 文件  
local: file1.txt remote: file1.txt  
200 PORT command successful. Consider using PASV.  
150 Ok to send data.  
226 File receive OK.  
6 bytes sent in 0.00 secs (266.3 kB/s)  
ftp> ls //查看 192.168.1.102 上/usr/project 目录的文件  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
-rw-r--r-- 1 1002 1003 6 Nov 15 13:16 file1.txt  
226 Directory send OK.  
ftp> delete file1.txt //删除 192.168.1.102 上/usr/project 目录下的 file1.txt 文件  
250 Delete operation successful. //删除成功  
ftp> ls //显示文件  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
226 Directory send OK.  
ftp> close //退出 soft1  
221 Goodbye.  
ftp> open 192.168.1.102 //连接 192.168.1.102  
Connected to 192.168.1.102.  
220 (vsFTPd 2.0.7)  
Name (192.168.1.102:dranyu): soft2 //以 soft2 身份登录  
331 Please specify the password.  
Password: //输入 soft2 用户的密码  
230 Login successful.
```

```

Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put file1.txt          //上传文件 file1.txt 到 192.168.1.102 上的
/usr/project 目录
local: file1.txt remote: file1.txt
200 PORT command successful. Consider using PASV.
553 Could not create file.    //上传文件失败
ftp> close                  //soft2 退出登录
221 Goodbye.
ftp> open 192.168.1.102      //连接 192.168.1.102
Connected to 192.168.1.102.
220 (vsFTPd 2.0.7)
Name (192.168.1.102:dranyu): soft3 //以 soft3 用户的身份登录
331 Please specify the password.
Password:                   //输入 soft3 用户的密码
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put file1.txt          //上传文件 file1.txt 到 192.168.1.102 上的 /usr/project 目录
local: file1.txt remote: file1.txt
200 PORT command successful. Consider using PASV.
553 Could not create file.    //上传文件失败
ftp> bye                    //退出 ftp
221 Goodbye.

```

16.3

电子邮件服务器 Postfix



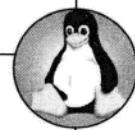
电子邮件是 Internet 应用服务之一，通过网络，用户可以以非常低廉的价格，非常快速的方式，与世界上任何一个网络用户联络。而电子邮件服务器是处理电子邮件的交换。它是为用户提供全由 E-mail 服务的电子邮件系统，人们通过访问服务器实现邮件的交换。

在 Ubuntu Linux 中，Postfix 是较为常见，而且是免费的邮件服务器。Postfix 试图成为一个快速、易于管理、安全的邮递程序，而且又能和 sendmail 有足够的兼容性，避免让原有的使用者感到不适。因此，Postfix 的外部运作如 sendmail，但实际上，内部是完全不同的设计。Postfix 被设计成在重负荷之下仍然可以正常工作。当系统运行超出了可用的内存或磁盘空间时，Postfix 会自动减少运行进程的数目。当处理的邮件数目增长时，Postfix 运行的进程不会跟着增加。

16.3.1 Postfix 简介

Postfix 是另一种 sendmail 的替代软件。Wietse Venema 到 IBM 的 T. J. Watson 研究中心做学术休假的一年里，启动了 Postfix 这个项目。Postfix 的设计目标包括：开放源代码的软件发布政策、快速性能、健壮、灵活和安全，同时还与 sendmail 保持足够的兼容性。目前，所有主要的 Linux 发行版本都包含 Postfix，Ubuntu 将 Postfix 软件包列为 main 软件库中的一个软件，这意味着它拥有安全更新。

Postfix 拥有以下特点：



- 免费

Postfix 想要作用的范围是广大的 Internet 用户，试图影响大多数的 Internet 上的电子邮件系统，因此它是免费的。

- 收发邮件快速

Postfix 在性能上大约比 sendmail 快三倍。一部运行 Postfix 的台式 PC 每天可以收发上百封邮件。

- 兼容性好：

Postfix 更着眼于作为 sendmail 的直接替换，使用 Postfix 替换 sendmail 却相当简单，因为 Postfix 使用的很多文件和 sendmail 一致，只需要在配置文件中指明原有 sendmail 配置文件的位置就可以了，Postfix 甚至还提供了 sendmail 和 mailq 程序，以保持兼容性。基本上，可以直接从 Sendmail 转换为使用 Postfix，而不需要额外的设置。Postfix 提供的安装配置方式也相当简单，它使用中心化的配置文件和非常易理解的配置指令。

- 更健壮

Postfix 被设计成在重负荷之下仍然可以正常工作。当系统运行超出了可用的内存或磁盘空间时，Postfix 会自动减少运行进程的数目。当处理的邮件数目增长时，Postfix 运行的进程不会跟着增加。Postfix 提供的一些强大的功能主要在于多种数据库表查询方式，例如它支持 DB、DBM、passwd 文件、正则表达式、MySQL 数据库，以及 LDAP 方式的查询，用于支持系统级的别名、虚拟主机等。虽然在 Qmail 中也能支持这些功能，但不如 Postfix 那样统一、简洁。此外，更改 Postfix 的设置之后，也不需要重新启动整个系统，只需要使用 Postfix reload 就能完全刷新配置，这也避免了丢失邮件的可能性。

- 更灵活：

Postfix 是由超过一沓的小程序组成的，每个程序完成特定的功能。你可以通过配置文件设置每个程序的运行参数。

- 安全性

Postfix 具有多层防御结构，可以有效地抵御恶意入侵者。如大多数的 Postfix 程序可以运行在较低的权限之下，不可以通过网络访问安全性相关的本地投递程序等。Postfix 考虑到了安全问题，它甚至不向 root 分发电子邮件，以避免以 root 身份读写文件或启动外部程序。所以 Postfix 的安全性是非常值得信赖的。

16.3.2 Postfix 的总体结构

Postfix 是模块化的应用程序，由几个小的协作程序所组成，这些程序负责发送网络消息、接收消息、在本地投递邮件等，它的系统结构如图 16-2 所示。

在 Ubuntu 中，master 程序启动和监视所有的 Postfix 进程。而 Postfix 的配置文件 master.cf 罗列出了 Postfix 的辅助程序，设置应该怎样启动它们的信息。在哪个文件中设定的默认值，适合于除了非常慢或者非常快的机器之外的所有情况使用。一般而言，系统管理员并不需要对它进行相关的调整。若系统管理员不需要启动某个应用程序，可在此配置文件中注释掉这一程序所在的行。

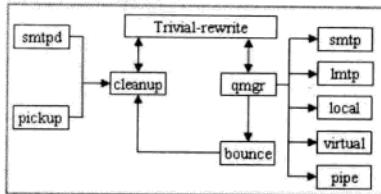


图 16-2 Postfix 的总体结构

1. 邮件队列

Postfix 应用程序中的 qmgr 进程控制着等候投递的邮件队列，这个队列管理器管理着如下 5 个邮件队列。

- incoming（传入）：此队列存放刚接收到的新邮件。
- active（活动）：此队列存放正在发送的邮件。
- deferred（推迟）：此队列存放发送失败的邮件。
- hold（约束）：此队列存放被系统管理员阻止发送的邮件。
- corrupt（错误）：此队列存放不可读或者不可分析的邮件。

队列管理器一般采用一种简单的 FIFO（先进先出）策略来选择要处理的下一个消息或邮件，同时它也支持一种复杂的抢先式算法，该算法优先发送收件人少的消息，然后再发送拥有多个收件人的邮件，对这些邮件进行批量发送。

2. 接收邮件

Postfix 应用程序中的 smtpd 进程负责在 SMTP 端口上收邮件。它会检查已连接的客户机是否得到发送邮件的授权。当通过/usr/lib/sendmail 这个 Postfix 兼容 sendmail 的程序往本地发邮件的时候，smtpd 进程会在/var/spool/postfix/maildrop 目录下写入一个邮件。而 pickup 程序则定期扫描这个目录，处理它发现的任何新文件。

然而，所有电子邮件都会由 cleanup 进程进行处理。cleanup 进程会根据 canonical 和 virtual 映射，给邮件加上缺少的信头，重写邮件地址。在电子邮件插入到 incoming 队列之前，cleanup 要用 trivial-rewrite 把它略做重写，这个程序对地址做一些小的修正，例如，给不完整的地址追加一个邮件域名。

3. 发送邮件

trivial-rewrite 进程帮助 qmgr 确定应该把邮件发到哪里。trivial-rewrite 做出的路由决策可以被 transport 映射所覆盖。

smtpd 进程负责通过 SMTP 协议把邮件发送到远程主机。lmtp 进程则使用 RFC 2022 制定的 LMTP 协议（Local Mail Transfer Protocol，本地邮件传送协议）发送邮件。LMTP 基于 SMTP，但这个协议已经做了修改，不要求邮件服务器管理邮件队列。

local 进程的工作是向本地发送电子邮件。它在别名表里解析电子邮件地址，按照从收件人的.forward 文件里找到的指示处理邮件。消息或邮件不是被转发给另一个地址传给一个外部程序处理，就是保存在该用户的邮件目录里。

virtual 进程把电子邮件发送到“虚拟邮箱”，也就是说，和本地的 Linux 账号没有关系的邮箱，但仍然代表有效的邮件目的地。最后，pipe 进程实现了通过外部程序发送邮件。



16.3.3 安装并运行 Postfix

Ubuntu 没有默认安装 Postfix 应用软件，系统管理员需要安装它才能实现邮件的收发功能。安装 Postfix 应用程序的命令如下所示：

```
root@dranyu-desktop:~# apt-get install postfix
```

执行上述命令之后，我们并不能使用 Postfix 服务器，还需要对其进行一番设置，执行如下命令并按照提示输入相关的信息：

```
root@dranyu-desktop:~#dpkg-reconfigure postfix
```

按下 Enter 键，系统会提示如下问题，我们输入“--”后面的内容来回答，如下所示：

```
General type of configuration? <-- Internet Site //设置接收网络上的邮件  
Where should mail for root go <-- NONE //不设置 root 身份的邮件的存贮位置，使用默认设置  
System mail name: <--mail.web3ds.org //设置邮件服务器的服务器名称  
Root and postmaster mail recipient: <-- /usr/mail //设置 root 身份的邮件收发保存的目录  
Other destinations to accept mail for? (blank for none) <-- dranyu, dranyu-desktop,  
localhost.localdomain, localhostt,mail.web3ds.org //限制所接收邮件的来源  
Force synchronous updates on mail queue? <-- No //设置不强制同步更新邮件队列  
Local networks? <--127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 //设置本地网络网段  
Mailbox size limit <-- 0 //设置不限制存放邮件目录的大小  
Local address extension character? <-- + //设置本地地址的扩展字符为+  
Internet protocols to use? <-- all //使用所有的网络协议
```

设置完毕后，命令的执行结果显示如下所示：

```
root@dranyu-desktop:/etc/postfix# dpkg-reconfigure postfix  
* Stopping Postfix Mail Transport Agent postfix [ OK ]  
setting synchronous mail queue updates: false  
changing /etc/mailname to mail.web3ds.org  
setting myorigin  
setting destinations: dranyu, dranyu-desktop, localhost.localdomain,  
localhostt,mail.web3ds.org  
setting relayhost:  
setting mynetworks: 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128  
setting mailbox_size_limit:  
setting recipient_delimiter: +  
setting inet_interfaces: all  
setting default_transport: smtp  
setting relay_transport: smtp  
setting inet_protocols: all  
WARNING: /etc/aliases exists, but does not have a root alias.
```

```
Postfix is now set up with the changes above. If you need to make changes, edit  
/etc/postfix/main.cf (and others) as needed. To view Postfix configuration  
values, see postconf(1).
```

```
After modifying main.cf, be sure to run '/etc/init.d/postfix reload'.
```

```
Running newaliases  
* Stopping Postfix Mail Transport Agent postfix [ OK ]  
* Starting Postfix Mail Transport Agent postfix [ OK ]
```

此时，Postfix 服务器已完全安装到了 Ubuntu 系统中了，系统管理员可正常使用 Postfix 应用程序的电子邮件收发功能。

为了测试 Postfix，我们安装一个包含 mail 命令的软件包 mailx，使用 mail 命令可快速查看邮件和发送邮件。安装 mailutils 软件包的命令如下所示：

```
root@dranyu-desktop:~# apt-get install mailx
```

安装完毕后，Postfix 应用程序自动启动。系统管理员也可通过如下命令启动 Postfix：

```
root@dranyu-desktop:~# /etc/init.d/postfix start
```

在 Postfix 的运行期间，系统管理员可使用如下命令停止它的运行：

```
root@dranyu-desktop:~# /etc/init.d/postfix stop
```

若系统管理员修改了 Postfix 的相应配置文件，可不用重启计算机系统，只要重启 Postfix 服务器即可，执行命令如下所示：

```
root@dranyu-desktop:~# /etc/init.d/postfix restart
```

16.3.4 配置 Postfix

Postfix 应用程序使用多个配置文件来完成对各个进程的管理。main.cf 文件是 Postfix 的首要配置文件。master.cf 文件配置服务器上各个功能模块的程序，它还定义了各种查找表，供 main.cf 访问，并且提供不同类型的服务映射。在使用 Postfix 之前，系统管理员必须知道配置文件中配置参数的含义，唯有这样才能在配置 Postfix 服务器时游刃有余。

1. 配置文件

在 Ubuntu 系统中安装的 Postfix 应用程序会将它的配置文件放置到/etc/postfix 目录下。这个目录下包括了其的主要配置文件 main.cf 和 master.cf。

在安装 Postfix 时，我们执行了 dpkg-reconfigure postfix 命令。这一命令将系统管理员所做的设置保存到主配置文件 main.cf 中，我们使用 Gedit 编辑器打开/etc/postfix/main.cf 文件，文件部分内容如下所示：

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version
...
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no
# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

myhostname = dranyu-desktop
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
```



```
mydestination      =      dranyu,      dranyu-desktop,      localhost.localdomain,
localhost,mail.web3ds.org
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
myorigin = /etc/mailname
default_transport = smtp
relay_transport = smtp
inet_protocols = all
```

main.cf 文件中的参数设置有好几百个，我们不可能一一将它们列出，在这里，我们简述一些比较常用的参数设置项的含义，如下所示。

- myhostname: 此项的值是此服务器的主机名。
- alias_maps: 此参数告诉 Postfix 在用户使用别名时应该用哪个数据库表。
- alias_database: 此项与 alias_maps 参数项类似，只是 alias_database 局限于由 Postfix 使用 newaliases 命令来编译所建立的表。
- mydestination: 此项参数告诉 Postfix 需要负责哪些网域的邮件。
- mynetworks: 此项参数声明受信任的网络。
- mailbox_size_limit: 此项参数的值指定 Postfix 使用的邮件空间的大小。
- inet_interfaces: 此项参数配置邮件的传送，控制着 Postfix 为其接收外来邮件的 IP 地址。
- myorigin: 此项参数用于发送邮件，向收件人提供明确的电子邮件的来源。
- default_transport: 此项参数显示了默认的邮件传输协议。
- relay_transport: 此项参数指定了默认的接收协议。
- inet_protocols: 此项参数指定 Postfix 支持的 IP 协议，有 3 种情况可供选择：all、ipv4 和 ipv6。

从配置文件，我们可知 Ubuntu 中的 Postfix 邮件服务器默认支持 SMTP。Main.cf 是 Postfix 的主要配置文件，系统管理员所需要的功能都可在此文件中进行相应的配置。

另一个主要配置文件是 master.cf，它管理着 Postfix 服务器中的所有进程的启动方式，以及进程启动所使用的参数。配置文件 master.cf 的部分内容如下所示：

```
1.=====
2 # service type private unpriv chroot wakeup maxproc command + args
3 #           (yes)   (yes)   (yes)  (never) (100)
4 # 5=====
5 smtp      inet  n   -   -   -   -   smtpd
6 ...
7 pickup    fifo  n   -   -   60    1   pickup
.....
```

master.cf 文件是以列表的结构组织进程的。上述的第二行就是表头，各字段的含义如下。

- 服务名称 service

服务器组件的名称。实际的命名规则，随该服务的传送类型（第二栏）而定。

- 传送方式 type

此字段表示传送服务所用的通信方法。有效的传送方式包括 `inet`、`UNIX` 与 `fifo`。`inet` 方法表示服务可通过“网络套接字”来访问，这类服务的对象可以是同系统上的其他进程，或是网络上其他主机的客户端进程。网络套接字服务的名称，用 `inet` 服务提供者的“IP 地址”与“通信端口”的组合来表示，或者是主机名称与`/etc/service` 定义的端口的符号名称组合，例如：192.168.1.2:25、`localhost:smtp`。如果服务方恰好位于本地主机上，则“IP 地址”与冒号都可以省略。

- 私有的 `private`

某些服务组件仅供 Postfix 系统自己使用，不开放给 Postfix 之外的其他软件使用。如果本栏标示为 `y`，表示私有访问（默认值）；`n` 代表开放公共访问。`inet` 类型的组件必须标示为 `n`，否则外界就无法访问该服务，毕竟网络套接字本身的用意，就是要开放给其他进程访问。

- 非特权的 `unpriv`

此字段指定是否使用非特权账户。默认值为 `y`，表示服务组件运行时，只需使用 `mail_owner` 参数指定的非特权账户（默认值为 Postfix），即以完成任务所需的最低限度权限来提供服务。大部分 Postfix 组件都可以使用非特权账户。对于需要 `root` 特权的服务组件，此栏必须设定为 `n`。

- 改变根目录 `chroot`

是否要改变组件的工作根目录，借此提升额外的安全性。工作根目录的位置由 `main.cf` 的 `queue_directory` 参数决定。此栏的默认值为 `y`（表示要改变工作根目录），大部分的 Postfix 组件也都可以在 `chroot` 环境下运作。不过，标准的安装方式是让所有组件都在正常环境下运行。将服务组件放在 `chroot` 环境下，添加了许多额外的复杂事情，你应该先通盘了解 `chroot` 所带来的保障。然后，再决定这样的额外安全性是否值得你多费一番设定与维护的工夫。关于 `chroot` 的问题，请参阅本章关于“`chroot`”的讨论。

- 唤醒间隔 `wakeup`

某些组件必须每隔一段时间被唤醒一次，定期执行它们的任务。`Pickup daemon` 就是这样一个例子，其默认休眠间隔是 60 秒，`master daemon` 每隔一分钟就唤醒 `Pickup` 一次，要求它检查 `maildrop` 队列是否收到新邮件。`qmgr` 和 `flush daemon` 也是需要被定期唤醒的服务组件。在时间值之后尾随一个问号（?），表示只有在需要该组件时才予以唤醒，0 表示不必唤醒。此字段的默认值为 0，因为目前只有 3 个组件需要被定期唤醒。Postfix 包预先为这 3 个组件设定的唤醒间隔时间，应该足以应付大部分情况，其他服务组件都不需要 `master` 的定期唤醒。

- 进程数上限 `maxproc`

此字段指定可以同时运行的进程个数的上限。如果没指定，则以 `main.cf` 的 `default_process_limit` 参数为准，其默认值为 100。如果设定为 0，表示没有任何限制。如果服务器系统的资源有限，或是想让系统在某方面的表现特别好，你可以调整 `maxproc` 的值。

- 命令 `command+ args`

最后一字段是运行服务的实际命令。命令中的“程序文件名”部分不必包含路径信息，因为 `master daemon` 假设所有程序文件都放在 `daemon_directory` 参数所指定的目录下，默认目录为`/usr/lib/postfix/`。Postfix 的所有程序皆提供“`-V`”选项，可用来提高日志信息的详细程度，当我们需要解决问题时，经常利用这种方法来获得更多、更有用的调试信息。此外，你可以



使用 -D 选项，让 Postfix 程序产生调试信息给调试程序。如果你需要知道更多关于调试的技术，请参考 Postfix 包随附的 DEBUG_README 文件。

2. 基本配置收发邮件

需要将一台主机设置为电子邮件服务器必须在 DNS 上添加相关的设置。16.1.4 节讲述了如何配置主域名服务器。为了构建电子邮件服务器，必须在正向解析数据库文件和反向解析数据库文件中添加邮件服务器的指向。配置基本收发邮件功能的具体操作步骤如下：

(1) 在正向解析数据库文件 db.web3ds.org 中添加如下代码行：

```
@ IN MX 10 mail.web3ds.org.  
mail IN A 192.168.1.106 //将 IP 地址为 192.168.1.106 的主机设置为邮件服务器
```

(2) 在反向解析数据库文件 db.192.168.1 中添加如下代码行：

```
106 IN PTR mail.web3ds.org.
```

(3) 在 Postfix 主配置文件 main.cf 中，必须包括如下内容：

```
mydestination = dranyu, dranyu-desktop, localhost.localdomain,  
localhost,mail.web3ds.org,web3ds.org  
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.1.0/24  
mailbox_size_limit = 0  
recipient_delimiter = +  
inet_interfaces = all  
myorigin = /etc/mailname  
default_transport = smtp  
relay_transport = smtp  
inet_protocols = all
```

在 mydestination 添加主机所在网域 web3ds.org 和主机域名 mail.web3ds.org。inet_interfaces 选项设置为 all，这样才可接收网络上的邮件。

(4) 配置完毕之后，重启 Postfix 服务器，执行命令如下所示：

```
root@dranyu-desktop:~# /etc/init.d/postfix restart  
* Stopping Postfix Mail Transport Agent postfix [ OK ]  
* Starting Postfix Mail Transport Agent postfix [ OK ]
```

16.3.5 测试 Postfix 服务器

配置好 Postfix 服务器的基本配置之后，Postfix 就可支持 SMTP 邮件收发。接下来，测试 Postfix 的邮件收发功能。

1. 修改 DNS 的解析服务器

目前，我们使用的是局域网，我们所设置的 DNS 服务器为 192.168.1.106，这样为了让发送邮件的进程可找到 mail 服务器，我们需要设置 DNS 的解析服务器。操作方法是使用 Gedit 文本编辑器打开/etc/resolv.conf 文件，将其内容修改为如下内容：

```
nameserver 192.168.1.106
```

这样，我们在主机就可识别 mail.web3ds.org 服务器。

2. mail 命令的使用

在测试之前，需要了解收发邮件的命令——mail。mail 命令是一个指令，同时也是一个 mail 程序。作为指令，它的基本语法如下：

```
mail [选项] mail_address
```

mail 命令常用的选项如下。

- **-i:** 此项表示忽略 tty 的中断讯号。
- **-I:** 强迫设成互动模式。
- **-v :** 此项表示列印出讯息，例如送信的地点、状态等。
- **-n :** 此项表示不读入 mail.rc 设定档。
- **-s :** 此项表示邮件标题。
- **-c cc_address:** 此项设置密抄邮件地址，cc_address 为密抄邮件地址列表。
- **-b bcc_address:** 此项设置隐藏密抄邮件地址，bcc_address 为隐藏密抄邮件地址列表。参数 mail_address 表示发件人的邮箱地址，如 dranyu@mail.web3ds.org。

在不使用任何的选项和参数时，使用命令 mail 就可进入 mail 应用程序提供的字符交互操作界面，在终端显示命令提示符如下：

```
dranyu@dranyu-desktop:~$ mail
"/var/mail/dranyu": 4 messages 4 new
& //&就是命令提示符
```

在该命令提示符下常用的命令解释如下。

- **unread :** 将指定的邮件标记为未读邮件。
- **h|headers :** 此项表示显示当前的邮件列表。
- **l|list :** 显示当前支持的命令列表。
- **?|help :** 显示多个查看邮件列表的命令参数用法。
- **d :** 删除当前邮件，指针并下移。如 d 1-100 表示删除第 1 到 100 封邮件。
- **f|from :** 只显示当前邮件的简要信息。如 f num 给显示某一个邮件的简要信息。
- **z :** 显示刚进行收件箱时的后面 20 封邮件列表。
- **more|p|page :** 阅读当前指针所在的邮件内容时，按空格键就是翻页，按回车键就是下移一行。
- **t|type|more|p|page num :** 阅读序列号为 num 的邮件。
- **n|next :** 阅读当前指针所在的下一封邮件内容。阅读时，按空格键表示翻页，按回车键表示下移一行。
- **v|visual :** 当前邮件进入纯文本编辑模式。
- **top :** 显示当前指针所在的邮件的邮件头。
- **file|folder :** 显示系统邮件所在的文件，以及邮件总数等信息。
- **x :** 退出 mail 字符操作界面，并不保存之前的操作，比如删除邮件。
- **q :** 退出 mail 字符操作界面，保存之前的操作，比如删除已用 d 删除的邮件，已阅读邮件会转存到当前用户目录下的 mbox 文件中。如果在 mbox 中删除文件才会彻底删除。
- **cd :** 改变当前所在文件夹的位置。
- **R num:** 回复邮件给序号为 num 邮件的发送人。
- **r num:** 回复邮件给序号为 num 邮件的发送人与其他接收人。
- **m:** 发送邮件。

在使用 mail 字符操作界面中，写信时，连续两次按“Ctrl+C”组合键则中断工作，不发送此信件；读完信，再按一次“Ctrl+C”组合键，退出阅读状态。



3. dranyu 发邮件到 root

使用 dranyu 的身份登录到终端，然后使用 mail 命令发送邮件到 root@mail.web3ds.org，命令执行过程和结果显示如图 16-3 所示。

```
dranyu@dranyu-desktop:~$ mail
No mail for dranyu
dranyu@dranyu-desktop:~$ mail to-addr root@mail.web3ds.org
Cc: root@mail.web3ds.org
Subject: mail1
This is mail1.
test.
admin
.
dranyu@dranyu-desktop:~$
```

图 16-3 发送邮件

补充说明

mail 命令在编写邮件时，以“.”为结束标记。当编写完邮件的内容后，在新的一行输入“.”，然后按下回车键即可。随即邮件就会被 Postfix 服务器处理，并发送到指定的邮箱地址。

4. root 回复 dranyu 的邮件

以 root 的身份登录终端，然后在终端输入 mail 命令，此时屏幕显示 root 的邮件列表，然后在提示符“&”后输入 1 就可阅读第 1 封邮件，执行命令结果如图 16-4 所示。

```
root@dranyu-desktop:~$ mail
root@dranyu-desktop:~$ mail
"/var/mail/root": 1 message 1 new
>N 1 dranyu          11月 22 09: 16/513  mail1
& 1
To: <to-addr@dranyu-desktop,<root@mail.web3ds.org>
Cc: <root@mail.web3ds.org>
Subject: mail1
Date: Sun, 22 Nov 2009 09:31:42 +0800 (CST)
From: dranyu@mail.web3ds.org (dranyu)

This is mail1.
test.
admin
&
```

图 16-4 阅读邮件

root 用户已接收到了 dranyu 用户发送的邮件。root 用户要回复此邮件，需要在提示符“&”后输入 r1，按下回车键开始编辑回复邮件。“r1”表示回复第 1 封邮件。邮件编写完毕后在末行输入“.”结束邮件的编写，并让 Postfix 发送回复邮件，如图 16-5 所示。

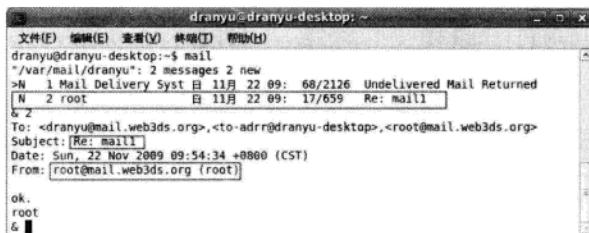
```
& r1
To: dranyu@mail.web3ds.org,to-addr@dranyu-desktop,root@mail.web3ds.org,<root@mail.web3ds.org>
Subject: Re: mail1

ok.
root
&
```

图 16-5 回复邮件

然后，再以 dranyu 的身份查看 root 的回复邮件，在终端上输入 mail 命令，在列表中找

到 root 的回复邮件 “Re:mail1”，如图 16-6 所示。



```
dranyu@dranyu-desktop:~$ mail
dranyu@dranyu-desktop:~$ mail
"/var/mail/dranyu": 2 messages 2 new
>N  1 Mail Delivery Syst 日 11月 22 09: 68/2126 Undelivered Mail Returned
N  2 root          日 11月 22 09: 17/659 Re: mail1
b 2
To: <dranyu@mail.web3ds.org>,<to-addr@dranyu-desktop>,<root@mail.web3ds.org>
Subject: [Re: mail1]
Date: Sun, 22 Nov 2009 09:54:34 +0800 (CST)
From: [root@mail.web3ds.org (root)]
ok.
root
b 1
```

图 16-6 阅读回复邮件

16.4 小结



本章介绍了网络应用服务器的构建，其中包括域名服务器、文件传输服务器和电子邮件服务器。

本章详细介绍了 Bind9 的安装过程，对 Bind9 服务器的配置文件进行了解说，并在主配置文件中配置了主域名服务器和辅域名服务器。两个完整的配置实例能帮助读者快速构建域名服务器。

本章使用 vsFTPD 文件传输服务器，将一台主机设置为一个开发项目的工作主机，这样开发项目中的成员都可按自己的权限访问主机。

在本章的最后描述如何配置 Postfix 电子邮件服务器，对 Postfix 的主配置文件中常用的参数进行了详述。

16.5 习题



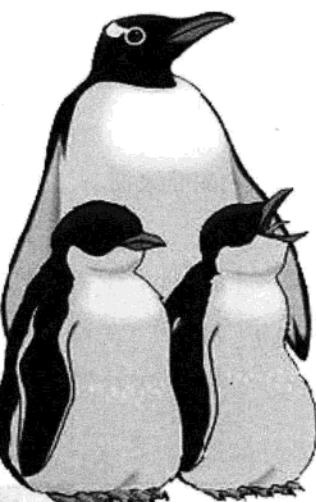
- 什么是域名系统？使用域名系统解决了什么问题？
- 在主机中安装 Bind9 服务器，完成基本的配置。
- Bind9 中的资源记录有多少种？它们的含义是什么？
- FTP 有几种用户类型？
- 在 vsFTPD 的配置文件中，如何设置禁止匿名登录？如何设置匿名用户具有对目录拥有可写的权限以及创建文件夹的权限？
- 在本机上将 vsFTPD 配置为一个组共享的服务器。
- 简述 Postfix 的特点。
- Postfix 的邮件队列管理器管理多少个邮件队列？这些队列的作用是什么？
- 在本机上完成 Postfix 的安装和配置，并测试 Postfix 收发邮件的功能。
- 使用 mail 命令读取 root 的邮件，并回复邮件。

Linux

第17章

文件服务器

目前，主流的操作系统不少，其中包括了 Windows、Linux 和 Mac 等。Windows 操作系统中提供了网络邻居的服务，可让局域网内的用户共享不同主机上的资源。Linux 系统之间同样拥有这样共享计算机资源的服务，那就是 Samba 和 NFS。而且通过 Samba 和 NFS，各个操作系统的资源可实现共享。这样，Linux 系统、Windows 系统和 mac 系统之间可互相访问系统资源。本章介绍 Samba 和 NFS 的安装和配置，并讲述如何在 Linux 中配置共享系统资源。





17.1 Samba 服务器概述

Samba 是向 Linux 提供资源共享的应用程序，其使用了 SMB（Server Message Block）通信协议实现系统中文件、目录和打印机的共享。在 Windows 系统中，NetBIOS 指定的就是 SMB。在 Linux 系统中，使用 Samba 就可使得 Linux 和 Windows 操作系统可以相互通信，达到网内不同操作系统的系统资源的共享，充分利用系统资源。

17.1.1 SMB 协议简介

SMB 最初是 IBM 的贝瑞·费根鲍姆（Barry Feigenbaum）研制的，其目的是将 DOS 操作系统中的本地文件接口“中断 13”改造为网络文件系统。后来微软对这个发展进行了重大更改，这个更改后的版本也是最常见的版本。微软将 SMB 协议与它和 3Com 一起发展的网络管理程序结合在一起，并在 Windows for Workgroups 和后来的 Windows 版本中不断加入新的功能。

SMB 一开始的设计是在 NetBIOS 协议上运行的，Windows 2000 引入了 SMB 直接在 TCP/IP 上运行的功能。目前，绝大多数的 PC 上都在运行这一协议，Windows 系统都充当着 SMB 协议的客户端和服务器。所以，SMB 是一个遵循客户机服务器模式的协议。SMB 服务器负责通过网络提供可用的共享资源给 SMB 客户机，服务器和客户机之间通过 TCP/IP 协议或者 IPX 协议或者是 NetBEUI 进行连接。一旦服务器和客户机之间建立了一个连接，客户机就可以通过向服务器发送命令完成共享操作，比如读、写、检索等。

SMB 协议尤其适用于局部子网，但是也可以被用来通过万维网链接不同的子网。SMB 协议为了屏蔽操作系统之间的差异采用了命名管道继承认证的机制。同时 SMB 对文件使用一种特殊的、被称为伺机锁定的锁定机构来提高文件传输速度。

目前，SMB 协议得到了广泛的应用，UNIX/Linux 系统都使用了这种通信协议，其中 Samba 应用程序就是使用 SMB 协议开放的，符合 GPL 的自由软件，其应用到 UNIX/Linux 系统中实现 SMB 协议，以实现不同系统之间的资源共享。

17.1.2 Samba 简介

Samba 是由澳大利亚人 Andrew Tridgell 于 1991 年为代替 PC-NFS 而开发的。后来项目发布到网络上，有许多程序员参与，从而产生了 Samba 开发小组。在 Samba 开发小组的共同努力下，Samba 已不仅仅存取及分享 SMB 的资料夹及打印机，本身还可以整合入 Windows Server 的网域、扮演为网域控制站（Domain Controller），以及加入 Active Directory 成员，从而为 Windows 与 UNIX 系列 OS 之间搭起一座桥梁，让两者的资源可互通有无。目前，Samba 最新版本是 3.4 版本，其的官方网站 <http://www.samba.org> 负责 Samba 的开发、维护和更新发布。

1. Samba 的组成进程

Samba 应用程序由多个进程组成，其中包括了 smbd、nmbd、smbclient、smbprint、testprns



和 testparm 等。这些进程负责不同的功能，有机地组合在一起，为 UNIX/Linux 的操作系统提供资源共享服务。在这些进程中，smbd 和 nmbd 进程是 samba 的核心守护进程，它们常驻系统内存中，管理着 Samba。Smbd 进程处理来自客户端的连接、处理文件、授权和用户名的工作；nmbd 进程负责帮助客户端找出服务器的位置，以进行浏览工作和管理域。Samba 是系统服务，其的 smbd 进程监听着 139/tcp 端口，nmbd 进程监听着 137 和 138/UDP 端口。

2. Samba 的主要功能

Samba 的功能在不断地完善和增强，以下是 Samba 提供的主要功能：

(1) 共享 Linux 的文件系统资源。Samba 让其他操作系统可访问 Linux 下的文件资源，使得访问 Linux 下资源如同访问本机资源一样简单。

(2) 共享 Samba 服务器上的打印机。当作为服务器的 Linux 安装了打印机，通过 Samba，其他系统的用户可使用此服务器上的打印机。

(3) 使用 Windows 系统的共享资源，如文件和打印机等。Linux 系统用户同样可通过 Samba 访问 Windows 系统的文件、目录和打印机。

(4) 支持主机的域名访问。Windows 可通过 Linux 主机的主机名来访问 Linux 主机，减去了使用 IP 地址访问的麻烦。

3. Samba 服务器与客户端的通信过程

Samba 应用程序是一种 c/s 结构的应用程序，这样其必然有一个通信过程，Samba 服务器与客户端的通信过程如图 17-1 所示。

(1) 客户端向 Samba 服务器发送一个 SMB negprot 请求数据报，并列出它所支持的所有 SMB 协议版本。Samba 服务器收到请求信息后，对 SMB negprot 请求数据报进行分析，并判断客户端使用的协议版本是否与服务器上提供的版本一致。如果没有一致的协议版本则返回 0xFFFFFH，结束通信。

(2) 协议确定后，客户端进程向服务器发起一个用户或共享的认证。客户端向 Samba 服务器发送 SesssetupX 请求数据报，即客户端发送一对用户名和密码或一个简单密码到服务器。若通过认证，Samba 服务器通过发送一个 SesssetupX 应答数据报来拒绝本次连接。若通过认证，Samba 服务器允许客户端的连接请求。

(3) 当客户端和服务器完成上述两个步骤的认证之后，客户端会发送一个 Tcon 或 TconX SMB 数据报并列出它想访问网络资源的名称或位置。Samba 服务器接收到数据报后，判断客户端请求的资源是否存在，若请求资源不存在，Samba 服务器会发送一个 TconX 应答数据报以表示此次连接被拒绝；否则发送一个 TconX 应答数据报以表示此次连接被允许，并返回相应的共享资源列表。

(4) 连接到相应资源后，SMB 客户端就能够通过 open SMB 打开一个文件，通过 read SMB 读取文件，通过 write SMB 写入文件，通过 close SMB 关闭文件。

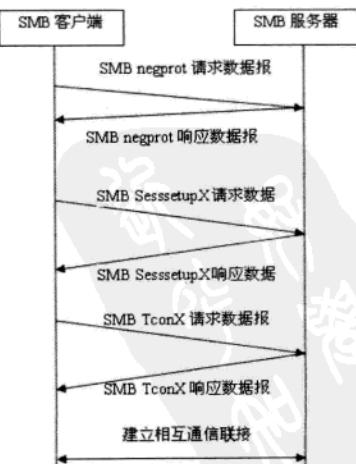


图 17-1 Samba 客户端与服务器通信连接过程

17.1.3 Samba 服务器安装和启动

在默认情况下，Ubuntu 并没有安装 Samba 应用程序，只安装了一个 Samba-common 的软件包，此 Samba-common 软件包的内容是 Samba 服务器和客户端都需要使用的一些函数库和配置文件等基本文件。我们可在终端的命令提示符后输入 Samba 命令，测试系统是否已安装 Samba 服务，命令执行和结果显示如下所示：

```
dranyu@dranyu-desktop:~$ smbd
程序 'smbd' 尚未安装。 您可以通过输入以下命令安装:
sudo apt-get install samba
bash: smbd: 找不到命令
```

smbd 是 Samba 的核心守护程序，它没有安装也就是说此时系统没有安装 Samba 服务。这时，我们需要安装 Samba 服务，执行命令和结果显示如下：

```
dranyu@dranyu-desktop:~$ sudo apt-get install samba smbclient //执行安装命令
[sudo] password for dranyu: //输入 dranyu 用户的密码，按下回车键
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
将会安装下列额外的软件包:
  samba-common
建议安装的软件包:
  openbsd-inetd inet-superserver smbldap-tools ldb-tools smbfs
下列【新】软件包将被安装:
  samba
下列的软件包将被升级:
  samba-common smbclient
.....
Importing account for nobody...ok
Importing account for dranyu...ok
Importing account for sunny...ok
Importing account for soft1...ok
Importing account for soft2...ok
Importing account for soft3...ok
* Starting Samba daemons [ OK ]
```

安装 Samba 和 smbclient 软件包后，并且已向 Samba 服务器中添加了系统当前的用户，系统自动启动了 Samba 服务。若在下次系统启动时，系统没有启动 Samba 服务，或是修改了 Samba 的配置文件，我们需要启动、停止和重启 Samba 服务，以及了解 Samba 的运行状态，使用命令如下：

- 启动 Samba 服务

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/samba start
* Starting Samba daemons [ OK ]
```

- 停止 Samba 服务

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/samba stop
* Stopping Samba daemons [ OK ]
```

- 重启 Samba 服务

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/samba restart
* Stopping Samba daemons [ OK ]
* Starting Samba daemons [ OK ]
```

- 查看 Samba 运行状态



```
root@dranyu-desktop:/home/dranyu# /etc/init.d/samba status
 * nmbd is running
 * smbd is running
```

17.2 Samba 的配置



在 Ubuntu 中，安装 Samba 服务后，Windows 用户可查看到 Ubuntu 的主机，可是不可访问，这时还需要对 Samba 进行一番配置。Ubuntu 提供了简单图形界面操作的共享方式，还提供了设置配置文件共享的方式。图形操作界面设置文件共享的方法会在 17.3 节讲述。这一节详细解说 Samba 的主配置文件 smb.conf。

17.2.1 smb.conf 配置文件的组成

Samba 服务的主配置文件 /etc/samba/smb.conf 由两部分构成，显示如下：

- Global Settings：该设置部分是全局设置，设置项都是与 Samba 服务整体运行环境有关的选项，它的设置项目是针对所有共享资源的。
- Share Definitions：该设置针对的是共享目录个别的设置，只对当前的共享资源起作用。

17.2.2 smb.conf 的编写格式和语法

smb.conf 配置文件以 “;” 和 “#” 注释行；以 “\” 符号为换行；以 “%” 标识变量；设置的每个设置项都以赋值的方式出现，使用 “=” 完成赋值；使用 “[资源名称]” 表示共享的资源名称。其中在 smb.conf 定义的变量如下所示：

```
%S=当前服务名（如果有的话）
%P=当前服务的根目录（如果有的话）
%u=当前服务的用户名（如果有的话）
%g=当前用户所在的主工作组
%U=当前对话的用户名
%G=当前对话的用户的主工作组
%H=当前服务的用户的 Home 目录
%v=Samba 服务的版本号
%h=运行 Samba 服务机器的主机名
%m=客户机的 NETBIOS 名称
%L=服务器的 NETBIOS 名称
%M=客户机的主机名
%N=NIS 服务器名
%p=NIS 服务的 Home 目录
%R=所采用的协议等级（值可以是 CORE, COREPLUS, LANMAN1, LANMAN2, NT1）
%d=当前服务进程的 ID
%a=客户机的结构（只能识别几项：Samba, WfWg, WinNT, Win95）
%I=客户机的 IP
%T=当前日期和时间
```

PDG

17.2.3 smb.conf 全局设置

在 smb.conf 配置文件中，全局设置的节名称是 “[global]”，而且为了方便阅读，smb.conf 包含如下的一行注释：

```
#----- Global Settings -----
```

这行注释与下行注释之间的设置为全局设置：

```
#----- Share Definitions -----
```

全局设置影响着 Samba 的运行，设置项繁多，这里简单讲述常用的设置项，如下所示：

- workgroup = WORKGROUP

此项定义该 Samba 服务器所在的工作组或域。

- server string = %h server (Samba, Ubuntu)

此项设定此主机的描述，当我们通过网络邻居访问的时候可以在备注里看到这个内容，而且还可以应用 Samba 设定的变量。%h 是变量，表示运行 Samba 服务机器的主机名。

- interfaces = 127.0.0.0/8 eth0

此项默认不被使用，它指定 Samba 服务使用的网域。如果你的 Linux 服务器有一张以上的网卡，同时你希望限制只有一个网络可用，你就可在此设置所需的网段。若设置的项目超过一个，必须使用 “,”、空格和 Tab 键来分隔。

- log file = /var/log/samba/log.%m

此项设置日志文件的保存位置。

- max log size = 1000

此项设置日志文件的大小，防止其文件内容过大而引起资源空间不足。此项值的单位是 KB。

- security = user

此项定义 Samba 的安全级别，默认不被启动，那么 Samba 执行 user 的安全级别。Samba 的安全级别从低到高分为四级：share，user，server，domain。它们对应的验证方式如下。

➤ share：没有安全性的级别，任何用户都可以不要用户名和口令访问服务器上的资源。

➤ user：Samba 的默认配置，要求用户在访问共享资源之前必须先提供用户名和密码进行验证。

➤ server：与 user 安全级别类似，但用户名和密码是递交到另外一个服务器去验证，比如递交给一台 NT 服务器。如果递交失败，就退到 user 安全级。

➤ domain：这个安全级别要求网络上存在一台 Windows 的主域控制器，Samba 把用户名和密码递交给它去验证。

后面三种安全级都要求用户在本 Linux 机器上也要系统账户。否则不能访问 Samba。

- encrypt passwords = true

此项设置是否使用加密过的口令。值为 true 表示使用，为 false 表示不使用。

- passdb backend = tdb

此项指定保存密码的文件格式是什么。此处设置为 tdb 格式。

- UNIX password sync = yes



此项设置是否同步 Windows 和 Linux 的口令。yes 表示同步，no 表示不同步。

- passwd program = /usr/bin/passwd %u

此项可以指定用来设置 UNIX 账号密码的程序，默认启动此项。其中%u 是变量，表示用户名。

- passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:* %n\n *password\supdated\ssuccessfully* .

此项可用来设置上网在进行 Linux 密码转换成 Samba 服务器密码时，屏幕出现的指示字符串，以及与用户进行交互的交互窗口。默认启动此项。

- map to guest = bad user

此项设置，当前用户登录失败时，或要求的连接的用户没有通过认证，那么系统将此用户映射为 guest 用户。

- load printers = yes

此项表示设定是否自动共享打印机而不使用设置下面的[printer]一节的相关内容。

- printing=bsd

此项定义打印系统的类型。打印系统的类型包括：bsd、sysv、plp、lprng、aix、hpx 和 qnx。

- include = /home/samba/etc/smb.conf.%m

此项允许 Samba 服务器使用其他的状态设置文件，这可方便管理员事先为不同的主机设计适合的设置文件，然后再以此方法套用到不同的主机。默认不启动此项。

- usershare allow guests = yes

此项指定共享文件的资源可匿名访问。若值为 no，表示不允许访问。

- domain master = yes|no

此项让 nmbd 成为一个域浏览器，取得各个本地主浏览器的浏览列表，并将整个域的浏览列表递交给各本地主浏览器。

17.2.4 smb.conf 共享设置

使用 Samba 服务时，设置权限等方面在 smb.conf 的全局部分进行设置，而要共享相关的资源则需要在 smb.conf 文件的共享设置部分进行设置。这样组织编写有利于阅读设置。smb.conf 中的共享设置部分从如下注释开始到文件结尾：

```
#----- Share Definitions -----#
```

在共享设置部分，包含了许多以中括号[]开头的区域，而且每个区域都代表了一个共享资源，是独立的，而且都有着相同的设置项。这些设置项如下所示：

- comment=""

此项设置共享的备注。

- browsable=yes|no

此项设置共享资源是否可浏览。

- path=

此项指定共享的路径，其内容可包含 Samba 的变量，如/home/dranyu、/home/%m。

- guest ok=yes | no

此项设置匿名用户的访问，值为 yes 表示允许匿名用户访问共享资源；值为 no 表示不允许。

- guest only=yes | no

此项设置是否只允许匿名用户访问。设置为 yes 表示只允许匿名用户访问，否则其他用户都可访问。

- read only=yes | no

此项指定了 path 目录默认是否可写。

- create mask=

此项设置访问文件的权限。例：0700 表示拥有者具有读、写和执行的权限，其他用户没有任何权限。一般不使用此项。

- directory mask

此项同 create mask 设置项类似，设置访问文件目录的权限。

- read list=

此项设置拥有只读权限的用户列表。如 read list=dranyu,sunny。

- write list=

此项设置拥有写权限的用户列表。如 write list=dranyu,sunny。

- valid users=

此项设置允许使用服务的用户列表。如 valid users=dranyu,sunny。

- invalid users=

此项设置不允许使用服务的用户列表。如 invalid users=dranyu,sunny。

- public=yes | no

此项设置被共享目录或设备是否允许网络中的匿名用户访问。如 public=yes 表示允许网络中的匿名用户访问。

在此部分设置共享资源的区域节如下所示：

```
[netlogon]
comment = Network Logon Service //注释：comment 表示备注信息
path = /home/samba/netlogon      //共享/home/samba/netlogon 目录资源
guest ok = yes                   //允许匿名用户登录或使用此共享资源
read only = yes                  //用户只有可读的权限
```



17.3 使用 Samba 共享实例

在此处我们设置共享一个/home/dranyu/doc 文件目录。

17.3.1 设置用户的 Samba 密码

Ubuntu 的账号需要使用 Smbpasswd 命令添加到 Samba 的密码文件中，才能使用 Samba 共享的资源。在 Samba 安装的时候，安装程序就将当时的用户信息复制到了 Samba 中，可并



没有设置密码。这样在默认的安全级别下，使用 Ubuntu 系统的账号是无法使用 Samba 的资源的。Smbpasswd 命令的基本语法如下：

```
smbpasswd [选项] 用户名
```

Smbpasswd 命令常用选项如下。

- **-a**：此选项表示增加 Samba 用户，并且同时设置新的口令。如果 Smbpasswd 文件中已经存在了这样的用户时，命令就变成通常的改口令模式。注意，所要加入的 SMB 用户必须是在系统口令文件/etc/passwd 中已经存在的用户，否则加入操作将会失败。只有 root 运行 Smbpasswd 程序时才可以使用这个选项。
- **-d**：此选项表示删除 Samba 用户，使指定用户不能访问 Samba 服务器。同样，也只有 root 运行 Smbpasswd 程序时才可以使用这个选项。

此处将 Ubuntu 的账号 dranyu 添加为 Samba 的用户，代码如下所示：

```
root@dranyu-desktop:/home/dranyu# smbpasswd -a dranyu  
New SMB password: //第一次输入密码，然后按下回车键  
Retype new SMB password: //第二次输入密码，然后按下回车键
```

其中-a 命令表示向 Samba 添加一个用户 dranyu，并设置 dranyu 的密码。

17.3.2 使用配置文件设置共享资源

在 Ubuntu 系统中，使用 Samba 设置共享资源的方法有两种，包括了设置配置文件和使用图形操作界面。设置配置文件的方法设置共享的步骤如下所示：

(1) 以 root 管理员的身份登录终端，命令如下所示：

```
dranyu@dranyu-desktop:/etc/samba$ sudo su  
[sudo] password for dranyu: //输入 dranyu 的密码
```

(2) 为了方便先对原有的配置文件进行必要的备份，备份文件名为 smb.conf.old，执行如下代码：

```
root@dranyu-desktop:/etc/samba# cp smb.conf smb.conf.old
```

(3) 编辑 smb.conf 配置文件，在 smb.conf 文件的最后位置添加如下代码：

```
[doc] //区域节名，同时也是共享资源的名称  
comment = //备注信息，此处没设置  
path = /home/dranyu/doc //共享资源的路径  
read only = yes //设置用户只有可读的权限  
guest ok = no //设置不允许匿名用户登录
```

(4) 保存 smb.conf 配置文件，使用 testparm 对 smb.conf 配置文件的编辑进行测试，执行命令与结果显示如下：

```
root@dranyu-desktop:/home/dranyu# testparm //测试 smb.conf 配置文件的语法  
Load smb config files from /etc/samba/smb.conf  
Processing section "[printers]"  
Processing section "[print$]"  
Processing section "[doc]"  
Loaded services file OK.  
Server role: ROLE_STANDALONE  
Press enter to see a dump of your service definitions  
//以下显示了 smb.conf 配置文件中所配置所有配置的简要信息  
[global]  
server string = %h server (Samba, Ubuntu)  
.....
```

```
[print$]
comment = Printer Drivers
path = /var/lib/samba/printers

[doc]
path = /home/dranyu/doc
```

(5) 第4步可发现配置文件smb.conf的编写没有错误。因此，重启Samba服务器使用smb.conf配置文件起作用。重启Samba服务器的命令如下所示：

```
root@dranyu-desktop:/etc/samba# /etc/init.d/samba restart
* Stopping Samba daemons
* Starting Samba daemons
[ OK ]
[ OK ]
```

通过上述步骤就实现/home/dranyu/doc文件目录的共享了。

17.3.3 使用图形操作界面设置共享

Ubuntu提供了使用图形操作界面设置资源共享的功能。使用图形操作界面共享/home/dranyu/doc文件目录的步骤如下所示：

- (1) 选择顶部面板“位置”|“主文件夹”菜单命令，进入dranyu的主目录，选中目录下的doc目录，然后用鼠标右键单击此文件目录，弹出菜单如图17-2所示。
- (2) 在快捷菜单中，选择“共享选项”选项，弹出File Manager对话框，选中“共享此目录”选项，其他参数保持不变，如图17-3所示。

补充说明

File Manager对话框中“允许他人在目录里写入数据”选项表示允许登录的用户从客户端中修改此目录里的内容。“注释”选项表示此目录在客户端下显示时提示的文本内容。而“允许匿名登录”选项表示用户在不输入账号或者没有账号的情况下同样可访问此目录。

- (3) 在File Manager对话框中，设置好相关的参数后，单击“创建共享”按钮，完成共享设置。

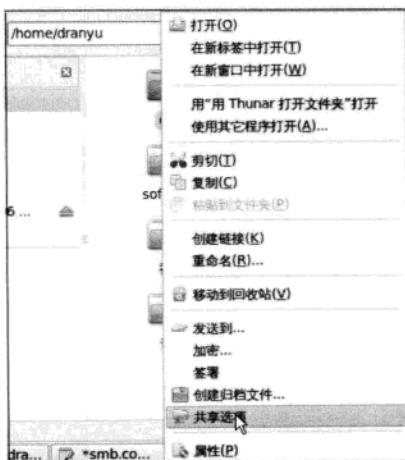


图 17-2 快捷菜单

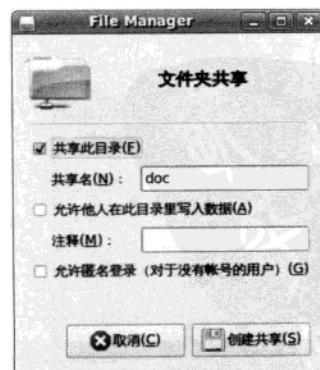


图 17-3 设置共享



17.3.4 使用 Samba 的客户端

Samba 服务器将资源共享了，那么在客户端如何访问这些共享资源呢？这就需要使用到 Samba 的客户端。Ubuntu 提供了多种 Samba 客户端的访问方法，此处将讲述以下两种方法：图形访问和 Smbclient 命令的字符操作界面访问。

1. 图形操作界面访问共享资源

使用图形操作界面访问共享资源的步骤：

- (1) 选择顶部面板“位置”|“主文件夹”菜单命令，进入 dranyu 的主目录，在“位置”输入框中输入 smb://192.168.1.106 然后按下回车键，结果显示如图 17-4 所示。



图 17-4 罗列 Samba 服务器的共享资源

其中 192.168.1.106 就是 Samba 服务器的 IP 地址。在罗列的资源中，doc 和 print\$文件目录就是 192.168.1.106 服务器上的共享资源。

- (2) 然后双击 doc 文件目录，弹出认证对话框，在“用户名”输入栏上输入 Samba 的用户名，如 dranyu；在“域”输入栏输入 Samba 服务器所在的域，如 WORKGROUP；在“密码”输入栏输入相应的用户密码；其他参数默认，如图 17-5 所示。

- (3) 单击“连接”按钮即可进入到共享资源 doc 文件目录，如图 17-6 所示。

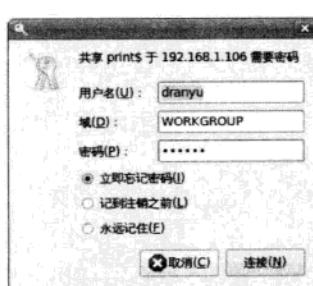


图 17-5 用户认证



图 17-6 192.168.1.106 服务器上的 doc 目录文件列表

补充说明

在不知道服务器的 IP 地址的时候，可选择顶部面板的“位置”|“网络”菜单命令，浏览整个内网的共享资源。使用这种文件就如同使用 Windows 系统上的“网络邻居”一样简单。或者在文件管理器左边的“位置”列表框中，选择“网络”选项，同样可实现对内网的共享资源的访问。

2. 字符操作界面访问共享资源

Samba 的客户端提供了一个类似 FTP 客户程序的 Samba 客户端程序 smbclient，通过此程序 Ubuntu Linux 可访问 Windows 共享或 Linux 提供的 Samba 服务。smbclient 的语法格式如下：

```
smbclient [网络资源] [密码] [-EhLN] [-B<IP 地址>] [-d<排错层级>] [-i<范围>] [-I<IP 地址>] [-l<记录文件>] [-M<NetBIOS 名称>] [-n<NetBIOS 名称>] [-O<连接槽选项>] [-p<TCP 连接端口>] [-R<名称解析顺序>] [-s<目录>] [-t<服务器字码>] [-T<tar 选项>] [-U<用户名>] [-W<工作群组>]
```

使用中括号 “[]” 包含的内容表示是可选选项，而使用尖括号 “<>” 包含的内容表示相对应选项的设置值。smbclient 命令的部分选项解析如下。

- [网络资源]：[网络资源]的格式为//服务器名称或 IP/资源分享名称，如 //192.168.1.106/share。
- [密码]：输入存取网络资源所需的密码。
- -B<IP 地址>：此项设置传送广播数据包时所用的 IP 地址。
- -d<排错层级>：此项指定记录文件所记载事件的详细程度。
- -E：此项指定将信息送到标准错误输出设备。
- -h：此项指定需要显示 smbclient 命令的帮助。
- -i<范围>：此项设置 NetBIOS 名称范围。
- -I<IP 地址>：此项指定服务器的 IP 地址。
- -l<记录文件>：此项指定记录文件的名称。
- -L：此项需要 smbclient 显示服务器端所分享出来的所有资源。
- -M<NetBIOS 名称>：可利用 WinPopup 协议，将信息送给选项中所指定的主机。
- -n<NetBIOS 名称>：指定用户端所要使用的 NetBIOS 名称。
- -N：使用此项表示使用匿名用户连接服务器。
- -O<连接槽选项>：设置用户端 TCP 连接槽的选项。
- -p<TCP 连接端口>：指定服务器端 TCP 连接端口编号。
- -R<名称解析顺序>：设置 NetBIOS 名称解析的顺序。
- -s<目录>：此项指定 smb.conf 所在的目录。
- -T<tar 选项>：备份服务器端分享的全部文件，并打包成 tar 格式的文件。
- -U<用户名>：指定用户名。
- -W<工作群组>：指定工作群组名称。

举例如下所示。

例 17-1：罗列 Ubuntu 下 Samba 服务器上的共享资源信息

罗列 Linux 操作系统下 Samba 服务提供的共享资源命令和执行结果显示如下所示：



```
root@dranyu-desktop:/etc/samba# smbclient -L 192.168.1.106 -U dranyu
//注释：192.168.1.106 是一个台安装了 ubuntu Linux 主机的 IP 地址，并且是 Samba 服务器，dranyu 是此 Samba 服务器的一个用户。
Enter dranyu's password:      //注释：在此提示下输入 dranyu 的 Samba 的密码
//注释：以下为命令执行结果
Domain=[DRANYU-DESKTOP] OS=[UNIX] Server=[Samba 3.3.2] //描述 192.168.1.106 主机
//此主机上的共享资源
  Sharename      Type      Comment
  -----
  print$        Disk       Printer Drivers
  doc           Disk       //显示共享资源的信息
  IPC$          IPC        IPC Service (dranyu-desktop server (Samba, Ubuntu))
  Samsung-SCX-4200 Printer   Samsung SCX-4200
Domain=[DRANYU-DESKTOP] OS=[UNIX] Server=[Samba 3.3.2]

//此主机所在网络的其他主机信息
  Workgroup      Master
  -----
  MSHOME         OUYANG   //此表示在此网络中有一台工作组为 MSHOME,
  主机名为 OUYANG 的主机
  WORKGROUP      DRANYU-DESKTOP
```

补充说明

在使用 smbclient 命令出现的术语包括 sharename、Type、Comment、Workgroup 和 Master。sharename 表示资源共享的资源名称；Type 表示此共享资源的设备类型，如 disk 类型表示为磁盘空间类型，Printer 表示打印机类型；Comment 表示备注；Workgroup 表示工作组；Master 表示主机的主机名。

例 17-2：匿名访问 Windows 系统主机的资源列表

以匿名用户的身份访问 Windows 所共享的资源列表，以下 192.168.1.105 所安装的系统是 Windows 系列操作系统，执行命令和结果显示如下所示：

```
root@dranyu-desktop:/etc/samba# smbclient -L 192.168.1.105 -N // -N 表示匿名登录，或不
使用密码登录
```

```
Domain=[DRANYU] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
```

Sharename	Type	Comment
-----	-----	-----
E\$	Disk	默认共享
IPC\$	IPC	远程 IPC
D\$	Disk	默认共享
print\$	Disk	打印机驱动程序
SharedDocs	Disk	
打印机	Printer	SnagIt 7
打印机 2	Printer	Microsoft XPS Document Writer
打印机 3	Printer	Microsoft Office Document Image Writer
F\$	Disk	默认共享
ADMIN\$	Disk	远程管理
H\$	Disk	默认共享
C\$	Disk	默认共享
software	Disk	

例 17-3：进入 smbclient 的字符操作界面访问 Linux

192.168.1.106 是 Linux 主机的 IP 地址, 其共享了一个文件目录名为 doc, 使用用户 dranyu 的身份登录进入 192.168.1.106 的 doc 目录, 并罗列此目录的内容, 代码如下:

```
root@dranyu-desktop:/etc/samba# smbclient //192.168.1.106/doc -U dranyu //输入命令,
按下
回车键
Enter dranyu's password: //输入 dranyu 用户的 samba 密码, 按下回车键
//进入 smb 的命令行提示操作模式
Domain=[DRANYU-DESKTOP] OS=[UNIX] Server=[Samba 3.3.2] //描述 samba 服务器的信息, OS 说明
此主机上所安装操作系统的类型, 此处为 UNIX 系列系统
smb: \> ls //罗列 192.168.1.106 共享资源 doc 目录下的文件
.
D 0 Mon Nov 23 11:07:56 2009
..
D 0 Thu Nov 26 21:14:38 2009
temple.txt 48 Sun May 31 00:31:26 2009

37487 blocks of size 131072, 12295 blocks available
smb: \> //smb 命令提示符
```

在 smb 命令提示符中输入 help 命令可查看命令操作界面上的命令及其使用方法, 如下所示:

```
smb: \> help //罗列命令行操作界面的使用命令
?
allinfo      altname      archive      blocksize
cancel       case_sensitive cd           chmod        chown
close        del          dir           du           echo
exit         get          getfacl      hardlink    help
history      iosize      lcd           link        lock
lowercase    ls           l              mask        md
mget         mkdir       more          mput        newer
open         posix       posix_encrypt posix_open  posix_mkdir
posix_rmdir  posix_unlink print        prompt     put
pwd          q            queue        quit        rd
recurse      reget      rename       reput      rm
rmdir        showacl     setmode     stat        symlink
tar          tarmode     translate   unlock     volume
vuid        wdel       logon       listconnect showconnect
!
smb: \> help get //列出 get 命令的使用方法, 此命令表示从 samba 服务器中获取指定的文件
HELP get:
<remote name> [local name] get a file
```

解释其中常用命令的含义如下。

- get: 从 Samba 服务器中获取指定文件, 并保存到本地主机上。
- ls 和 dir: 罗列出当前所进入的 Samba 服务器下的文件目录里的文件列表。
- put: 将文件从本地主机上传到 Samba 服务器的指定文件目录里。
- exit 和 quit: 表示退出 Samba 字符操作界面, 返回到终端。
- cd: 进入到指定的 Samba 服务器的目录下, 如同 Linux 下的 cd 命令。

17.4 NFS 文件服务器

网络文件系统 (NFS) 是一种在网络上类 UNIX 系统的主机间共享文件的方法, 通过这



种方法，访问远程的文件就如同访问本地硬盘上的文件一样。Ubuntu 既可以是 NFS 服务器也可以是 NFS 客户，这意味着它可以把文件系统导出给其他系统，也可以挂载从其他机器上导入的文件系统。

17.4.1 NFS 服务器简介

NFS 是分布式计算机系统的一个组成部分，可实现在异构网络上共享和装配远程文件系统。它由 SUN 公司于 20 世纪 80 年代所开发。目前，已经成为文件服务的一种标准(RFC1904, RFC1813)。NFS 运行在 SUN 的 RPC (Remote Procedure Call, 远程过程调用) 基础上, RPC 定义了一种与系统无关的方法来实现进程间通信。由此, NFS server 也可以看做是 RPC server。

NFS 具有以下的优点：

(1) 本地主机可以使用更少的磁盘空间，因为常用数据可以被保存在一台机器上，并让网络上的其他机器访问。

(2) 使用 NFS 可使 Linux 的用户信息保存到 NFS 服务器上，这样就不需要为用户在每台网络机器上存放一个用户目录。

(3) NFS 服务器上的存储设备如软盘、光驱及 USB 设备都可以被网络上其他机器使用。

使用 NFS 服务器网络管理员可集中管理网络的资源，从而减小企业中硬件资源的投入，提高资源的使用率。

17.4.2 安装和启动 NFS 服务器

在 Ubuntu 系统中，NFS 服务器被默认安装。系统管理员可使用新立得软件包管理器安装 nfs-kernel-server 软件包，若是在终端上输入如下命令：

```
root@dranyu-desktop:/home/dranyu# apt-get install nfs-kernel-server
```

NFS 服务器安装完毕之后，在终端显示的配置结果如下所示：

```
正在设置 portmap (6.0-9ubuntul) ... //注释：启动 portmap 守护进程
 * Starting portmap daemon... [ OK ]
 ----
 Creating config file /etc/default/nfs-common with new version
 正在添加系统用户 'statd' (UID 116)...
 正在将新用户 'statd' (UID 116) 添加到组'nogroup'...
 正在设置 nfs-kernel-server (1:1.1.4-1ubuntul) ...
 Creating config file /etc/exports with new version //注释：配置 NFS 的默认配置文件
 Creating config file /etc/default/nfs-kernel-server with new version
 //启动 NFS 服务器的守护进程，即启动 NFS 服务
 * Starting NFS common utilities [ OK ]
 * Exporting directories for NFS kernel daemon... [ OK ]
 * Starting NFS kernel daemon [ OK ]
```

从安装的显示结果可知，NFS 的服务器已经启动。使用 NFS 服务器，必须知道 NFS 服务器的启动、停止和重启等的管理功能，这些管理功能如下所示：

- 显示 nfs 服务器命令/nfs-kernel-server 的用法

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/nfs-kernel-server
 * Usage: nfs-kernel-server {start|stop|status|reload|force-reload|restart}
```

- 查看 nfs 服务器的运行状态

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/nfs-kernel-server status
```

```
nfsd running
```

- 停止 nfs 服务器的运行

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/nfs-kernel-server stop
 * Stopping NFS kernel daemon [ OK ]
 * Unexporting directories for NFS kernel daemon... [ OK ]
```

- 启动 nfs 服务器

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/nfs-kernel-server start
 * Exporting directories for NFS kernel daemon... [ OK ]
 * Starting NFS kernel daemon [ OK ]
```

- 重启 nfs 服务器

```
root@dranyu-desktop:/home/dranyu# /etc/init.d/nfs-kernel-server restart
 * Stopping NFS kernel daemon [ OK ]
 * Unexporting directories for NFS kernel daemon... [ OK ]
 * Exporting directories for NFS kernel daemon... [ OK ]
 * Starting NFS kernel daemon [ OK ]
```

17.4.3 配置 NFS 服务器

NFS 服务器安装之后，并没设置默认共享的资源，需要系统管理员对 NFS 服务器的配置文件进行相关的配置。NFS 的相关配置都集中在/etc(exports) 配置文件中。这一配置文件的内容控制着资源的存取，同时控制着客户端对共享资源的访问权限。默认的/etc(exports) 配置文件如下所示：

```
# /etc(exports: the access control list for filesystems which may be exported
#           to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
```

配置文件中每一行都使用“#”。以“#”开始的行表示注释，不会被应用到 nfs 服务器中。在此文件中，每一项记录代表一个共享资源，并设置了共享资源的存取权限，其语法格式如下：

共享资源的目录 [客户端(通常选项，用户 ID 映射)]

其中各选项的含义如下所示：

- 共享资源的目录

此字段指定共享资源的目录，必须使用绝对路径。如/home/dranyu、/和/mnt/sda 等。

- 客户端

此字段指定允许连接到此 NFS 服务器的客户端。此字段的表示方式有 4 种，如表 17-1 所示。

表 17-1 客户端表示方法

表示方法	说 明
单一主机	指定单一主机的域名、别名或 IP 地址。如果指定超过一个主机，必须使用空格分隔开如 dranyu.web3ds.org



续表

表示方法	说 明
群组	使用“@组名”的格式指定允许访问 NFS 服务器的群组，如@workgroup 表示允许从属于 workgroup 群组网络的主机访问 NFS 服务器
通配符	可使用“*”或“?”通配符指定允许访问 NFS 服务器的客户端。如*表示允许任何主机访问，*.web3ds.org 表示此允许从属于 web3ds.org 域的所有主机访问
IP 网络区段	此字段的格式为 IP 地址与网络掩码的组合，即 address/netmask，用来指定从属于此 IP 网络区段的主机可访问 NFS 服务器。如 192.168.1.0/255.255.255.0 或 192.168.1.0/24

● 通常选项

在/etc(exports 配置文件中使用的选项有很多，其中部分常用选项如表 17-2 所示。

表 17-2 exports 配置文件中的通用选项

选 项	说 明
rw ro	设置共享资源的存取权限。rw 表示允许客户端读写共享资源，相反 ro 只允许客户端读取共享资源
secure insecure	限定客户端的网络端口号的范围。Secure 表示要求客户端的网络端口号小于 1024，此为默认值；insecure 表示对网络端口号不作要求
sync async	设置 NFS 服务器写磁盘的方式。Sync 表示设置 NFS 服务器与客户端的写操作同步；使用 async 表示异步写
no_subtree_check subtree_check	设置文件系统与文件系统下的文件目录的访问方式。no_subtree_check 表示不使用 subtree 的安全检测机制，相反 subtree_check 表示使用 subtree 检测机制。在只共享文件系统中的一个文件目录时，必须添加 no_subtree_check 选项

● 用户 ID 映射

我们都知道 root 是系统管理员，它的 UID 是 0。NFS 服务器将资源共享，若不对连接到 NFS 服务器的用户 ID 做个映射就会容易造成安全漏洞。如若客户端的用户以 root 的身份连接到 NFS 服务器，如果 NFS 没有设置 ID 映射，这样客户端的 root 用户就拥有 NFS 服务器的系统管理员的权限，这样是不安全的。所以，在设置 NFS 时有必要设置连接用户 ID 的映射。这种映射方式就是将拥有高权限的用户 ID 映射为一般的用户 ID。如将 root 用户的 ID 映射为匿名用户 ID。此处的可用选项如表 17-3 所示。

表 17-3 用户 ID 映射

选 项	说 明
all_squash	共享文件的 UID 和 GID 映射匿名用户 anonymous，适合公用目录
no_all_squash	保留共享文件的 UID 和 GID（默认）
root_squash	root 用户的所有请求映射成如 anonymous 用户一样的权限（默认）
no_root_squash	root 用户具有根目录的完全管理访问权限
anonuid	指定匿名用户的 id。若 NFS 服务器上有一普通用户 dranyu，其的 UID 为 1005，设置 anonuid=1005 表示用户 ID 映射时用户的 ID 都映射到 1005，则对共享资源拥有 dranyu 用户的访问权
anongid	指定连接用户所映射群组的 GID

/etc(exports 配置文件的使用举例如下所示。

例 17-4： / dranyu (rw) sunny (rw,no_root_squash)

“/”表示系统的根文件系统。此例子表示名为 dranyu 的主机对 NFS 服务器的 “/” 拥有读写权限；而名为 sunny 主机对 NFS 服务器的 “/” 拥有读写权限，并且 “no_root_squash” 表示 sunny 主机上的 root 用户拥有对此目录的所有权限。

例 17-5: /home/dranyu/doc proj*.web3ds.org (rw,no_subtree_check)

proj*.web3ds.org 表示属于 web3ds.org 网域的以 proj 开头命名的所有主机都允许访问 NFS 服务器上的/home/dranyu/doc 文件目录，“no_subtree_check” 表示在访问时不检测它的父目录的权限。

例 17-6: /share 192.168.1.0/24(ro)

此例表示 192.168.1.0/24 网域上的所有主机对 NFS 服务器上的/share 文件系统拥有只读权限。

例 17-7: /home/sunny sunny2 (rw,all_squash,anonuid=1003,anongid=1003)

此例表示从主机名为 sunny2 连接到 NFS 的用户都将映射为群组 ID 为 1003 且用户 ID 为 1003 的 NFS 服务器上的用户，并且拥有对共享资源的读写权限。

例 17-8: /home/project @project (rw)

此例表示属于工作组 project 的所有主机对 NFS 服务器上的/home/project 拥有读写权限。

只要 NFS 服务器启动了，在/etc(exports 配置文件中输入共享资源目录之后并不需要重启 NFS 服务器，只需使用 exportfs 命令来检查/etc(exports 配置文件语法是否正确，将重新导出共享资源目录。exportfs 命令的基本语法如下：

`exportfs [-aruv] [客户端挂载的目录路径]`

其中参数说明如下。

- `-a`: 此选项表示全部挂载(或卸载) /etc(exports 文件内的共享资源目录)。
- `-r`: 重新挂载/etc(exports 里共享资源目录的设定，此外，亦同步更新/etc(exports 及 /var/lib/nfs/xtab 的内容。
- `-u`: 卸载某一目录。
- `-v`: 在 export 的时候，将共享资源目录显示到屏幕上。

配置 NFS 共享的具体操作步骤如下：

(1) 将/etc(exports 配置文件的共享资源目录内容如下所示：

```
/home/dranyu/doc *(rw,sync,no_subtree_check)
/ 192.168.1.0/24(rw,sync,no_subtree_check,all_squash)
/usr *.web3ds.org(ro,no_subtree_check)
/home/dranyu dranyu-desktop(rw,no_subtree_check)
```

(2) 设置好/etc(exports 配置文件后，保存配置文件的内容，然后检查 exports 配置文件并重新加载共享资源目录，执行命令和显示结果如下所示：

```
root@dranyu-desktop:/home# exportfs -rv //注释：使用-r 选项重新设定，-v 选项显示结果
exporting dranyu-desktop:/home/dranyu
exporting 192.168.1.0/24:/
exporting *.web3ds.org:/usr
exporting *:/home/dranyu/doc
```

17.4.4 客户端 NFS 的连接

配置好 NFS 服务器后，网络上已经拥有了共享资源，客户端并没使用到这些共享资源，



这需要在客户端中将共享资源挂载到客户端。使用客户端时，需要安装 nfs 的通用程序包，执行命令如下：

```
root@dranyu-desktop:/home# apt-get install nfs-common
```

安装此软件包后，就可使用 NFS 服务器的共享资源了。

1. 察看 NFS 服务器上的共享资源

NFS 客户端中可使用 showmount 查看指定的 NFS 服务器上的共享资源。showmount 命令的基本语法如下。

```
showmount [-ae] 域名或 IP 地址
```

其选项的含义如下：

- -a：此选项表示显示主机上有哪些客户端在使用共享资源。
- -e：此选项表示显示某主机已共享的资源。

例 17-9：查看 192.168.1.106 服务器上的共享资源

```
dranyu@dranyu-desktop:~$ showmount -e 192.168.1.106
Export list for 192.168.1.106:
/home/dranyu/doc *
/usr          *.web3ds.org
/             192.168.1.0/24
/home/dranyu  dranyu-desktop
```

例 17-10：查看使用着 192.168.1.106NFS 服务器上共享资源的客户端

```
root@dranyu-desktop:/home# showmount -a 192.168.1.106
All mount points on 192.168.1.106:
192.168.1.115:/home/dranyu/doc //注释：表示 IP 地址为 192.168.1.115 的主机使用着 NFS 服
务器上/home/dranyu/doc 的共享资源
```

2. 手动挂载 NFS 文件系统

使用 showmount 命令得知 NFS 服务器上共享的资源后，客户端需要挂载共享资源才可使用 NFS 服务器上的共享资源。挂载 NFS 共享资源的文件有多种，此处说明使用常用的挂载工具 mount；在下一小节中说明系统自动挂载的方法。

使用挂载工具 mount 挂载 NFS 服务器上的共享资源，如同挂载本地的磁盘空间资源，具体的使用方法在第 13.5.4 节中已做详述。在此使用 mount 的一个简单格式如下：

```
mount IP 地址或域名:共享资源目录 挂载点
```

其中，IP 地址或域名表示 NFS 服务器的 IP 地址或域名；共享资源目录表示 NFS 服务器上的共享资源的绝对路径；挂载点是客户端的一个空目录，作为共享资源的进入目录。

挂载一个 NFS 共享资源的具体操作步骤如下：

(1) 创建一个空目录。在客户端的/home 目录中创建一个名为 nfs 的目录，命令如下：

```
root@dranyu-desktop:/home# mkdir nfs
```

(2) 使用 mount 命令，将共享资源挂载到 nfs 目录下。执行命令如下：

```
root@dranyu-desktop:/home# mount 192.168.1.106:/home/dranyu/doc /home/nfs
```

(3) 使用共享资源。只要有足够的权限，管理 nfs 目录内的内容就如同管理客户端本身的文件内容一样。一些管理操作如下：

```
root@dranyu-desktop:/home# cd nfs //进入 nfs 目录
```

```
root@dranyu-desktop:/home/nfs# ls //罗列 nfs 目录的文件内容
```

```
temple.txt //显示 NFS 服务器中/home/dranyu/doc 的目录内容
```

补充说明

可使用 `showmount` 命令查看是否挂载成功。若客户端不需要挂载 NFS 服务器的共享资源，可使用 `umount` 卸载指定的共享资源目录，如此处 `umount nfs` 表示卸载 `nfs` 目录，即断开与 NFS 共享资源/`/home/dranyu/doc` 的连接。

3. 设置自动挂载 NFS 文件系统

除了使用 `mount` 挂载工具挂载 NFS 服务器上的共享资源，系统管理员还设置让客户端在系统启动的时候就挂载 NFS 共享资源，自动挂载 192.168.1.106NFS 服务器上的 `/home/dranyu/doc` 共享资源，具体的操作方法如下：

(1) 将`/etc/fstab` 配置文件进行备份，使用如下命令：

```
root@dranyu-desktop:~# cp /etc/fstab /etc/fstab.old
```

(2) 使用 Gedit 等编辑工具打开`/etc/fstab`，在 `fstab` 配置文件中添加如下格式的内容：

```
192.168.1.106:/home/dranyu/doc /home/nfs nfs rsize=8192, wsize=8192, 0 2
```

其中第一个字段 `192.168.1.106:/home/dranyu/doc` 表示 192.168.1.106NFS 服务器上的 `/home/dranyu/doc` 目录；第二个字段`/home/nfs` 表示客户端的挂载点目录；`nfs` 表示挂载来源于 NFS 服务器；`rsize` 表示读出的速度为 8192 字节；`wsize` 表示写入的速度为 8192 字节。

(3) 保存`/etc/fstab` 配置文件，然后重启计算机即可。

17.5 小结



本章向读者讲述了如何配置 Samba 服务器和 NFS 服务器，并详细解说了 Samba 配置文件和 NFS 配置文件的编写格式。在文中，配置了 Samba 服务器并实现了从客户端进行访问，描述了在客户端如何使用 Samba 共享的资源。

在本章最后，在已创建 NFS 服务器的网络环境下，说明了在客户端使用 NFS 服务器上的共享资源的方法，将共享资源设置为客户端在启动时自动加载的系统资源。

通过这章的学习，读者能够自行配置 Samba 服务器和 NFS 服务器，实现 Windows 系统与 Linux 系统的互访，以及同类 UNIX 系统之间的互访。

17.6 习题



1. Samba 软件包主要使用了什么协议？它的组成进程有哪些？
2. 简述 Samba 的主要功能。简述 Samba 服务器与客户端的通信过程。
3. 在终端上，使用 `apt-get` 命令安装 Samba 软件包。
4. Ubuntu 中如何启动、停止和重启 Samba 服务器？
5. 简述 Samba 主要配置文件 `smb.conf` 的组成部分与共享目录的编写格式。
6. 编辑`/etc/smb.conf`，将`/home` 里的主文件夹共享，允许匿名登录，允许其他用户拥有只



读权限。

7. 使用图形操作界面共享文件夹。
8. 如何使用 `smbclient` 连接到 191.168.1.106 下的一个名为 `doc` 共享文件目录？
9. 简述 NFS 文件系统的优点。
10. 安装 NFS 服务器，完成服务器的停止、启动和重启操作。
11. 使用`/etc/exports` 配置文件，共享 NFS 服务器上的`/home` 目录的资源，并设置所有用户都映射为匿名用户，连接的用户拥有读写权限。
12. 在客户端安装 NFS 客户端程序 `nfs-common`，完成察看 NFS 服务器相应共享资源的操作。
13. 使用挂载工具 `mount` 挂载 NFS 服务器的共享资源`/home`。
14. 将 NFS 服务器上的共享资源`/home` 添加到`/etc/fstab` 配置文件中，使得 NFS 服务器上的`/home` 能自动加载。

Linux

第18章

Web服务器

近年来，随着网络技术的普及、应用和 Web 技术的不断完善，Web 服务已经发展成为互联网上重要的服务形式之一。原来的客户端/服务器模式正在逐渐被浏览器/服务器模式所取代。

Web 服务器也称为 WWW (World Wide Web) 服务器，它的主要功能就是提供网上信息浏览服务。目前社会上使用最多的 Web 服务器包括：Apache 服务器，Tomcat 服务器，以及 Microsoft 的 IIS (Internet Information Server) 服务器等。其中 Apache 服务器是全世界使用人数最多的 Web 服务器。本章主要讲述 Apache 的安装和配置。





18.1 Web 服务器简介

因特网的出现改变了信息传播的速度和途径，使得信息传播越来越快，传播的途径越显多样性。Web 服务器的地位越显得重要。Web 服务器实现了信息的传播，让世界各地的人们都可查看到所需的信息。

18.1.1 Apache 服务器

1. Apache 的历史

Apache 是一种完全开放源码的 HTTP 服务器，可以在 Windows、Linux 和 UNIX 等大多数计算机操作系统中运行。由于它的多平台和安全性被广泛使用，是目前最流行的 Web 服务器端软件之一。并且几乎所有的 Linux 版本都捆绑了 Apache 服务器。

它最初的思想和源码来自于 NCSA (University of Illinois, Urbana-Champaign) 所开发的 NCSA httpd 1.3。在 1994 年 6 月，许多 Web 主管自行发展自己的外加功能及修正码。一小批的 Web 主管通过私人电子邮件聚集在一起实现他们的改变（以“patches”，即补丁的形式出现）。在 1995 年 2 月底，8 位核心贡献者成立原始的 Apache 组织，在 1995 年 4 月，Apache 0.6.2 版发布，这是第一个公开的版本。

从此以后，Apache 服务器就不断地发展，在原有特性的基础上，开发出了许多新的特性。1995 年 5 月到 7 月，一个新的服务器架构被开发出来（程序名为 Shambhala），包含了模块结构及 API。这个服务器架构是基于存储池的存储器配置及要调整的预设分支处理模式开发的。这个开发群体在 7 月将此服务器架构转换到新的服务器上，并为 Apache 0.7.x 新增特色，于 8 月推出了 Apache 0.8.8。在一年之内，Apache 服务器超过了 NCSA 的 httpd 成为 Internet 上排名第一的服务器。

为什么选择 Apache 这个名字？到目前为止，还没有一个确切的说法。但是，目前流传最广的也是比较可信的一个解释是：1995 年初，Apache 最早开发的时候，是由当时最流行的 HTTP 服务器 NCSA HTTPD 1.3 的代码修改而成的，因此它是“一个修补的 (a patchy)”服务器。然而，在服务器官方网站的 FAQ 中是这么解释的：“‘Apache’这个名字是为了纪念名为 Apache 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”。无论是哪种解释，Apache 2.x 分支不包含任何 NCSA 的代码。

Apache 的成功之处主要在于它的源代码开放、有一支开放的开发队伍、支持跨平台的应用，以及它的可移植性等方面。Apache 的强项是极大的市场占有率。Apache 是今天网际网络金字招牌第一号，远远将竞争者抛在后面。特别是，Apache 市场占有率达到几个使对手望尘莫及的优势。

- 起源于 HTTP 协议——降低了用户加入协议来支援新的应用软件的门槛。
- 给 UNIX/Linux 带来生机：Apache 走到哪里，UNIX/Linux 就走到哪里。
- 支援厂商的支持，为 Apache 提供的工具/模块持续成长。

经过一大批自主开发人员的不懈努力，Apache 服务器最终发展成为目前最流行的 Web

服务器端软件之一，世界上许多著名的站点（例如 Yahho, IBM, Red Hat 等）都是 Apache 的产物。

2. Apache 2.2 新特性

本书用到的 Apache 版本是 2.2，因此下面就介绍一下 Apache2.2 版本所具有的新特性。和 Apache 版本相比，它的新特性主要表现在以下几个方面。

- 认证/授权：原本捆绑在一起的认证与授权模块现在被分开了。新的 mod_authn_alias 模块极大地简化某些身份认证的配置。
- 配置：默认的配置布局已经被简化并模块化了，启用常用特性的配置片段现在已经和 Apache 捆绑在一起，可以被轻易地添加到主配置文件中。
- 代理：新增的 mod_proxy_balancer 模块为 mod_proxy 提供了负载均衡服务。新增的 mod_proxy_ajp 模块为 Apache Tomcat 使用的 Apache JServ Protocol version 1.3 提供了支持。
- 更新了正则表达式库：5.0 版的 Perl 兼容正则表达式库已经被包含进来了，通过 --with-pcre 编译选项，httpd 现在可以使用系统中已经安装好的 PCRE。
- Event MPM：event 多路处理模块（MPM）使用一个单独隔开的线程处理持久连接（Keep Alive）。传统上，持久连接要求 httpd 专门拿出一个工作者（worker）（也就是一个进程/线程）来处理它。这个专用的工作者在持久连接超时前不能被重新使用。
- SQL 数据库支持：mod_dbd 和 apr_dbd 框架（framework）一起为需要使用数据库的模块提供直接的支持，在线程化的 MPM 中还能支持连接缓冲池。

18.1.2 Tomcat 服务器

1. Tomcat 简介

Tomcat 服务器是一个免费的开放源代码的 Web 应用服务器，它是 Apache 软件基金会（Apache Software Foundation）的 Jakarta 中的一个核心项目，由 Apache、Sun 和其他一些公司及个人共同开发而成。由于有了 Sun 的参与和支持，最新的 Servlet 和 JSP 规范总是能在 Tomcat 中得到体现，例如，Tomcat 5 支持 Servlet 2.4 和 JSP 2.0 规范，Tomcat 6 实现了 servlet 2.5 和 JSP2.1 规范。因为 Tomcat 技术先进、性能稳定，而且免费，因而深受 Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的 Web 应用服务器。

在 Tomcat 中，应用程序的部署很简单，你只需将你的 WAR 放到 Tomcat 的 webapp 目录下，Tomcat 会自动检测到这个文件，并将其解压。在浏览器中访问这个应用的 Jsp 页面时，一般情况下第一次会很慢，因为 Tomcat 要将 Jsp 转化为 Servlet 文件，然后进行编译。编译以后，访问将会很快。另外，Tomcat 也提供了一个应用：manager，访问这个应用需要用户名和密码，用户名和密码存储在一个 xml 文件中。通过这个应用，辅助于 FTP，你可以在远程通过 Web 部署和撤销应用。

2. Tomcat 响应客户请求的过程

在一般情况下，如果客户端发出请求，Tomcat 服务器会按照下面的几个步骤对客户请求做出响应：

- 客户端（通常都是浏览器）访问 Web 服务器，发送 HTTP 请求。



- Web 服务器接收到请求后，传递给 Servlet 容器。
- Servlet 容器加载 Servlet，产生 Servlet 实例后，向其传递表示请求和响应的对象。
- Servlet 实例使用请求对象得到客户端的请求信息，然后进行相应的处理。
- Servlet 实例将处理结果通过响应对象发送回客户端，容器负责确保响应正确送出，同时将控制返回给 Web 服务器。

3. Tomcat 6.0 新特性

目前，Tomcat 最新的版本是 6.0x，和以前版本相比，它主要在以下几个方面进行了改进：

- Tomcat 6.0 的目录结构发生了很大的改变，废弃了原来常用的 common 目录，原来 common/lib 下的所有 jar 包现在都位于 lib 目录下。
- 安装 Tomcat 6.0 时，Java 虚拟机版本必须在 5.0 或者以上。
- 在 Tomcat 6.0 中，日志的配置独立于容器之外，因为原来的 commons-logging 被 org.apache.juli.logging 所代替了。
- 优化了 http IO connector 的性能，能更好地支持 Etag。
- 增加了 http NIO connector，使得大量并发网站的性能可以显著得到提高。
- 完全重构了 cluster。

18.1.3 其他 Web 服务器

1. IBM WebSphere

WebSphere 软件平台能够帮助客户在 Web 上创建自己的业务或将自己的业务扩展到 Web 上，为客户提供了一个可靠、可扩展、跨平台的解决方案。作为 IBM 电子商务应用框架的一个关键组成部分，WebSphere 软件平台为客户提供了一个使其能够充分利用 Internet 的集成解决方案。

WebSphere 软件平台提供了一整套全面的集成电子商务软件解决方案。作为一种基于行业标准的平台，它拥有足够的灵活性，能够适应市场的波动和商业目标的变化。它能够创建、部署、管理、扩展出强大、可移植、与众不同的电子商务应用，所有这些内容在必要时都可以与现有的传统应用实现集成。以这一稳固的平台为基础，客户可以将不同的 IT 环境集成在一起，从而能够最大限度地利用现有的投资。

WebSphere Application Server 是一种功能完善、开放的 Web 应用程序服务器，是 IBM 电子商务计划的核心部分，它是基于 Java 的应用环境，用于建立、部署和管理 Internet 和 Internet Web 应用程序。这一整套产品进行了扩展，以适应 Web 应用程序服务器的需要，范围从简单到高级直到企业级。

WebSphere 针对以 Web 为中心的开发人员，他们都是在基本 HTTP 服务器和 CGI 编程技术上成长起来的。IBM 将提供 WebSphere 产品系列，通过提供综合资源、可重复使用的组件、功能强大并易于使用的工具，以及支持 HTTP 和 IIOP 通信的可伸缩运行的环境，来帮助这些用户从简单的 Web 应用程序转移到电子商务世界。

2. Microsoft IIS

Microsoft 的 Web 服务器产品为 Internet Information Server (IIS)，IIS 是允许在公共

Intranet 或 Internet 上发布信息的 Web 服务器。IIS 是目前最流行的 Web 服务器产品之一，很多著名的网站都是建立在 IIS 的平台上。IIS 提供了一个图形界面的管理工具，称为 Internet 服务管理器，可用于监视配置和控制 Internet 服务。

IIS 是一种 Web 服务组件，其中包括 Web 服务器、FTP 服务器、NNTP 服务器和 SMTP 服务器，分别用于网页浏览、文件传输、新闻服务和邮件发送等方面，它使得在网络（包括互连网和局域网）上发布信息成了一件很容易的事。它提供 ISAPI（Intranet Server API）作为扩展 Web 服务器功能的编程接口。同时，它还提供一个 Internet 数据库连接器，可以实现对数据库的查询和更新。

18.2 构建 Apache 服务器



Ubuntu 9.04 安装光盘中已经自带了 Apache 软件包，其版本是 2.2 版。也可以到 Apache 网站下载其他版本，其官方网址为 <http://httpd.apache.org>。

本节我们将重点讲述一下在 Ubuntu Linux 操作系统下，如何安装 Apache 服务器，如何配置 Apache 服务器，如何配置模块和相关服务。

18.2.1 安装 Apache 服务器

目前，几乎所有的 Linux 发行版都内置了 Apache 服务器，Ubuntu 也不例外。但是，在默认情况下，Ubuntu Linux 安装程序不会将 Apache 服务器安装在系统上。众所周知，Ubuntu Linux 安装方式可以分为两种：图形界面安装和字符界面安装。在图形界面下安装 Apache 服务器比较简单，这里就不进行讲述了。下面我们就重点讲述一下在字符界面下安装 Apache 服务器的具体步骤。

第一步：在 Ubuntu Linux 操作系统下，打开一个终端，并且输入命令 `sudo apt-get install apache2`。

第二步：启动 Apache 服务器。我们可以用命令 `/etc/init.d/apache2 start` 来启动 Apache 服务器。

第三步：测试 Apache 服务器是否安装成功。在浏览器中输入网址 `http://localhost` 或者 `http://127.0.0.1`，如果页面显示“*It works!*”，则表明 Apache 服务器安装成功。

补充说明

此时 Apache 服务器上的网站内容保存在 `/var/www` 目录下。显示“*It works!*”的页面就保存在 `/var/www` 目录里。

18.2.2 Apache 配置文件

在 Windows 操作系统下，Apache 服务器的配置文件通常只有一个，就是 `httpd.conf`。但是，在 Ubuntu 9.04 版本下安装过 Apache 服务器的用户就会发现，它的 `httpd.conf`（位于 `/etc/apache2` 目录）是空的。而且还可以发现 Ubuntu 的 Apache 软件包的配置文件并不像



Windows 的那样简单，它把各个设置项分在了不同的配置文件中。简单说就是，在 Ubuntu Linux 下，Apache 服务器的配置文件不止一个，各个配置文件的功能也不一样。

和其他 Linux 版本不同的是，Ubuntu Linux 版本下 Apache 服务器的安装目录是 /etc/apache2，在该目录中包括几个文件：apache2.conf、conf.d、httpd.conf、mods-enabled、sites-available、apache2.conf~、envvars、mods-available、ports.conf、sites-enabled。下面就介绍一下各个文件的作用。

- apache2.conf 文件

apache2.conf 文件是最重要的，Apache 在启动时会自动读取这个文件的配置信息，并且其他的一些配置文件，如 httpd.conf 等，则是通过 `Include` 指令包含进来的。apache2.conf 配置文件中的部分内容如下：

```
ServerRoot "/etc/apache2" // 服务器主配置文件的目录，即服务器的根目录
ServerName 127.0.0.1 // 服务器的域名设置
PidFile ${APACHE_PID_FILE} // 指定保存 apache 服务器运行时的进程 ID，其中变量
${APACHE_PID_FILE} 的值在 envvars 配置文件中设置
Timeout 300 // 若在指定时间内没有收到或发出任何数据则断开与客户端的连接，单位为秒
KeepAlive On // 将 KeepAlive 选项设置为 On 可提高访问性能，其允许保持连接，最好让每次连接能提出多个请求，避免请求一个文件，就跟服务器建立一次连接
MaxKeepAliveRequests 100 // 使用 MaxKeepAliveRequests 选项可设置 apache 服务器的连接数，若设置为 0 表示无限制
KeepAliveTimeout 15 // 持续作用于服务器在两次请求之间的等待时间
User ${APACHE_RUN_USER} // 设置能启动 apache 服务器的用户，变量 ${APACHE_RUN_USER} 的值在 envvars 配置文件中设置
Group ${APACHE_RUN_GROUP} // 设置能启动 Apache 服务器的用户组，变量 ${APACHE_RUN_GROUP} 的值在 envvars 配置文件中设置
AccessFileName .htaccess // 设置保护目录配置文件的名称
<Files ~ "\.ht"> // 设置以 .ht 开头的访问权限
    Order allow,deny // 设置优先，allow 优先于 deny。allow 表示允许客户端访问，deny 表示禁止
    Deny from all // 禁止所有客户端访问
</Files> // File 语句的结束标签
DefaultType text/plain // 指定默认的 MIME 文件类型为纯文本 text 或 HTML 格式的文件 plain
HostnameLookups Off // 设置记录客户端 IP 地址，若值为 On 则表示记录客户端的主机名称
ErrorLog /var/log/apache2/error.log // 指定存放错误日志的位置
LogLevel warn // 设置错误日志的级别为 warn
# Include module configuration: // 指定 apache 的模块配置文件的搜索位置
Include /etc/apache2/mods-enabled/*.load // apache 模块的加载设置文件
Include /etc/apache2/mods-enabled/*.conf // apache 模块的配置文件
# Include all the user configurations: // 指定用户设置 apache 服务器的配置文件
Include /etc/apache2/httpd.conf
# Include ports listing // 指定 apache 服务器连接的端口配置文件
Include /etc/apache2/ports.conf
// 以下 5 行定义了 5 种记录日志的格式，分别名为 vhost_combined、combined、common、referer 和 agent
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
vhost_combined
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
// 使用上面所定义的日志格式记录连接到 Apache 的连接信息
CustomLog /var/log/apache2/other_vhosts_access.log vhost_combined // 使用
vhost_combined 格式记录日志
```

```
# Include the virtual host configurations: //指定虚拟主机服务器的配置文件
Include /etc/apache2/sites-enabled/
```

根据注释，可以很清楚地知道每个配置文件的大致作用。当然，也可以完全把所有的配置都放在 apache2.conf 或者 httpd.conf 或者任何一个配置文件中。

- envvars 文件

此文件指定可运行 Apache 服务器的用户和用户组，以及 Apache 运行时保存进程 ID 的文件。其文件内容如下：

```
export APACHE_RUN_USER=www-data //设置用户列表变量 APACHE_RUN_USER 的值为 www-data，并将
变量设置为环境变量
export APACHE_RUN_GROUP=www-data //设置用户组列表变量 APACHE_RUN_GROUP 的值为 www-data，并将
变量设置为环境变量
export APACHE_PID_FILE=/var/run/apache2.pid //设置保存 Apache 服务器进程 ID 的文件，值为
/var/run/apache2.pid，并将变量设置为环境变量
```

- httpd.conf 文件

httpd.conf 是用户配置文件。在默认情况下，它是一个空文件，在这里可以加入本机的一些虚拟主机都用到的全局配置。

- conf.d 文件夹

conf.d 文件夹的用途是存放一些通用的配置文件。

- ports.conf 文件

ports.conf 文件里面设置了 Apache 服务器使用的端口。如果需要调整默认的端口设置，建议编辑这个文件。

```
NameVirtualHost *:80 //设置虚拟主机通信接口
Listen 80 //设置 Apache 服务器的监听端口

<IfModule mod_ssl.c> //设置使用 ssl 模块时使用的 ssl 连接端口
# SSL name based virtual hosts are not yet supported, therefore no
# NameVirtualHost statement here
Listen 443
</IfModule>
```

- mods-available 和 mods-enabled 目录

mods-available 目录存放着当前这个 Apache2 所支持的所有模块的加载配置；而 mods-enabled 目录下存放着 mods-available 中相应的软链接。mods-enabled 的作用就像一个开关，需要加载某个模块时，从 mods-available 目录做一个软链接过来就可以了；不需要的话，可以删掉这个软链接，而模块配置还是存在的。

- sites-available 和 sites-enabled 目录

sites-available 和 sites-enabled 两个目录的作用是存放虚拟主机。和 mods-available、mods-enabled 一样，sites-available 目录是存放配置文件的，sites-enabled 目录保存软链接。

补充说明

Ubuntu 中 Apache 服务器的配置细分到每个配置文件。如端口的设置保存在 port.conf 配置文件中；虚拟主机的配置保存在 sites-available 目录中的 default 配置文件中等。这种将配置内容分到不同的配置文件中，可使 Apache 服务器的配置更加简单，配置内容更加明了。



18.2.3 Apache 的基本配置

Ubuntu 里 Apache 配置分布到各个配置文件。在网络管理员需要使用 Apache 服务器相应功能时，只需找到相关的配置文件并修改此配置文件即可达到设置的目的。Apache 服务器配置选项很多，本节讲述 Apache 服务器常用的配置。

1. 设置 Apache 服务器端口

设置 Apache 服务器的端口号是很有必要的。在一台拥有多个 Web 服务器或其他网络服务的服务器上，端口冲突是常见的事情。修改 Apache 服务器的端口号涉及两个配置文件，包括 port.conf 和 sites-available/default。

例 18-1：将默认的 80 端口号修改为 8088 端口，具体操作步骤如下所示。

(1) 使用 Gedit 文本编辑器打开/etc/apache2/port.conf，将其中原有的两行代码修改为如下代码：

```
NameVirtualHost *:8088  
Listen 8088
```

(2) 使用 Gedit 文本编辑器打开/etc/apache2/sites-available/default，将其中原有的一行代码修改为如下代码：

```
<VirtualHost *:8088>
```

(3) 保存已修改的配置文件。重启 Apache 服务器，命令执行和结果显示如下所示：

```
//重启 apache 服务器的命令  
root@dranyu-desktop:/etc/apache2/mods-enabled# /etc/init.d/apache2 restart  
* Restarting web server apache2  
... waiting [ OK ]
```

注意

上述修改的三行代码的端口号要保持一致，修改配置文件后一定要重启 Apache 服务器。

2. 设置网站的根目录

Apache 服务器提供发布信息的功能，它的网站资源保存在 Apache 服务器配置上所指定的网站根目录里。默认的网站根目录是/var/www 目录。在浏览器的地址栏上输入网络地址 http://localhost 时，Apache 服务器在/var/www 目录下搜索网站的首页，找到后将其解释为超文本发送到客户端的浏览器上。设置网站资源保存的根目录，可方便网站的发布并可使用安全机制保护网站。

例 18-2：将默认的网站根目录/var/www 修改为/home/dranyu/www 目录，具体操作步骤如下所示。

(1) 使用 Gedit 文本编辑器打开/etc/apache2/sites-available/default 文件，找到如下代码：

```
DocumentRoot /var/www //DocumentRoot 选项设置 Apache 服务器存放网站资源的根目录  
<Directory /var/www/> //<Directory>与</Directory>所包含的内容设置网络用户对指定文件  
目录的访问权限。此处设置/var/www/被访问的权限  
Options Indexes FollowSymLinks MultiViews //Options 选项用于定义目录使用哪些特  
性，这里设置 3 个值。Indexes 表示允许目录浏览，以 DirectoryIndex 列表中指定的首页文件优先浏览；  
FollowSymLinks 表示可以在该目录中使用符号连接；MultiViews 表示允许内容协商的多重视图。  
AllowOverride None //AllowOverride 选项用于定义位于每个目录下.htaccess（访问控制）  
文件中的指令类型。基于安全和效率的考虑，一般将此值设置为 None，表示禁止使用.htaccess 文件，而将目录权限的设置放在配置文件<Directory>和</Directory>语句之间。
```

```
Order allow,deny //Order 选项用于定义默认的访问权限 allow 和 deny 语句的处理顺序。此  
处值为 allow,deny 表示默认禁止所有客户机的访问，且 allow 语句在 deny 语句之前被匹配  
allow from all //allow 选项设置允许访问的客户机，all 值表示允许所有客户机访问。此处  
的值还可指定允许访问的主机 IP 地址，如 192.168.1.*。  
</Directory> //Directory 语句的结束
```

(2) 将上述代码中出现的 /var/www 字符串修改为 /home/dranyu/www，然后保存配置文件。

(3) 重启 Apache 服务器，网站的根目录即修改为 /home/dranyu/www。

补充说明

控制访问文件 .htaccess 保存着访问权限的规则。DirectoryIndex 列表的值在 /etc/apache2/mods-available/dir.conf 配置文件中设置。

3. 设置启动目录的起始页

通常我们在地址栏上输入一个没有指定文件的地址，浏览器还可找到网站的首页并显示此首页。这是由于 Web 服务器会自动搜索目录下的首页文件。此首页文件的名称是可设置的。

例 18-3：在 Ubuntu 下的 Apache 服务器中，设置浏览器搜索首页的具体步骤如下所示。

(1) 使用 Gedit 打开 /etc/apache2/mods-available/dir.conf 文件，其内容如下：

```
<IfModule mod_dir.c> //mod_dir.c 模块设置搜索的首页文件  
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm  
</IfModule>
```

(2) 在 DirectoryIndex 选项后添加希望 Apache 服务器搜索的首页名称，如 index.jsp、start.html 或 main.html 等。使用空格将每个文件名隔开。添加 index.jsp，并将 index.php 放置在第一位，修改后的代码如下：

```
<IfModule mod_dir.c> //mod_dir.c 模块设置搜索的首页文件  
    DirectoryIndex index.jsp index.php index.html index.cgi index.pl index.xhtml index.htm  
index.jsp  
</IfModule>
```

(3) 重启 Apache 服务器即可。

4. 设置网站别名

若一台主机上保存不止一个网站的资源，可使用网站别名来访问将要访问网站的资源。

例 18-4：将一个网站的根目录 web3ds 设置可以 http://localhost/web3ds 的方式访问其的资源，其具体步骤如下所示。

(1) 使用 Gedit 文本编辑器打开 /etc/apache2/mods-available/alias.conf，其的内容如下所示：

```
<IfModule alias_module> //<IfModule></IfModule>语句是 Apache 配置文件中的对模块的定义，此  
处设置 alias_module 模块  
    Alias /icons/ "/usr/share/apache2/icons/" //设置/usr/share/apache2/icons 文件目录中资源  
的访问别名  
    <Directory "/usr/share/apache2/icons"> //设置/usr/share/apache2/icons 文件目录的访问权  
限  
        Options Indexes MultiViews  
        AllowOverride None  
        Order allow,deny  
        Allow from all  
</Directory>  
</IfModule>
```

(2) 在 <IfModule> 和 </IfModule> 语句之间添加如下代码，将保存在 /home/dranyu/web3ds/



目录设置别名为/web3ds/。

```
Alias /web3ds/ "/home/dranyu/web3ds/"  
<Directory "/home/dranyu/web3ds "> //设置/usr/share/apache2/icons文件目录的访问权限  
    Options Indexes MultiViews  
    AllowOverride None  
    Order allow,deny  
    Allow from all  
</Directory>
```

(3) 重启 Apache 服务器后，在浏览器的地址栏上输入 <http://localhost/web3ds> 即可访问 /home/dranyu/web3ds 文件目录里的资源。



18.3 管理 Apache 服务器

Ubuntu 下管理 Apache 服务器使用的是/etc/init.d/apache2 脚本。这个脚本的用法与 shell 命令的用法一致，基本语法如下：

```
/etc/init.d/apache2 [选项]
```

其中常用选项如下所示。

- start：启动 Apache 服务器。
- stop：停止 Apache 服务器。
- restart：重启 Apache 服务器。
- status：显示 Apache 服务器的运行状态。
- reload：重启加载配置文件。

例 18-5：Apache 安装完毕之后，Apache 服务器处于运行状态，为说明 Apache2 脚本的使用，以及保持操作的连续性，在终端同时完成 Apache 服务器的停止、启动、显示状态、重启、重新加载配置文件等操作，操作过程和执行命令结果如图 18-1 所示。

```
root@dranyu-desktop:/etc/apache2/mods-enabled# /etc/init.d/apache2 stop  
* Stopping web server apache2  
... waiting [ OK ]  
root@dranyu-desktop:/etc/apache2/mods-enabled# /etc/init.d/apache2 start  
* Starting web server apache2  
root@dranyu-desktop:/etc/apache2/mods-enabled# /etc/init.d/apache2 status  
* Apache is running (pid 16810).  
root@dranyu-desktop:/etc/apache2/mods-enabled# /etc/init.d/apache2 restart  
* Restarting web server apache2  
... waiting [ OK ]  
root@dranyu-desktop:/etc/apache2/mods-enabled# /etc/init.d/apache2 status  
* Apache is running (pid 16914).  
root@dranyu-desktop:/etc/apache2/mods-enabled# /etc/init.d/apache2 reload  
* Reloading web server config apache2  
root@dranyu-desktop:/etc/apache2/mods-enabled# /etc/init.d/apache2 status  
* Apache is running (pid 16914).  
root@dranyu-desktop:/etc/apache2/mods-enabled#
```

图 18-1 管理 Apache 服务器

补充说明

使用 Apache2 脚本时，若不记得它的选项可输入不带参数的脚本。这样，终端会显示 Apache2 脚本的相关选项。

18.4 Apache 日志管理



日志是 Apache 服务器的重要组成部分。要有效地管理 Web 服务器，就有必要反馈服务器的活动、性能，以及出现的问题。Apache HTTP 服务器提供了非常全面而灵活的日志记录功能。

Apache 的标准中规定了 4 类日志：访问日志、错误日志、传输日志和 Cookie 日志。其中后两种被 Apache2.0 视为过时，即在 Ubuntu 9.04 版本的 Apache 中只存在访问日志和错误日志。

18.4.1 日志相关配置指令

Apache 2.2.*版本中有 4 条与日志相关的配置指令如表 18-1 所示。

表 18-1 日志相关配置指令

指 令	格 式	说 明
ErrorLog	ErrorLog 错误日志文件名	指定错误日志保存的路径
LogLevel	LogLevel 错误日志记录等级	指定错误日志的记录等级
LogFormat	LogFormat “记录格式说明串”格式昵称	为一个日志记录格式命名
CustomLog	格式 1: CustomLog“访问日志文件名” 记录格式说明串 格式昵称 格式 2: CustomLog “ 管道程序名访 问日志文件名”记录格式说明串 格式 昵称	指定访问日志保存的路径和记录格式；指定访问日志由指定的程序生成并指定日志记录格式
HostNameLookups	HostNameLookups On[Double]Off	设置 Apache 查找主机的名称或 IP

在默认安装中，Apache 的访问日志保存在 /var/log/apache2/access.log 文件中；错误日志保存在 /var/log/apache2/error.log 文件中。在 /etc/apache2/sites-available/default 配置文件中可找到访问日志和错误日志的配置代码如下所示：

```
ErrorLog /var/log/apache2/error.log //指定保存错误日志的文件 error.log
LogLevel warn //指定错误日志的日志记录的级别为 warn
```

```
.....
CustomLog /var/log/apache2/access.log combined //设置保存访问日志的文件 access.log,
以
及保存日志记录的格式
```

18.4.2 访问日志

通过查看访问日志可了解到客户端的访问信息。访问日志记录了该服务器所有的请求过程，主要记录的是客户的信息，通过它可以知道谁访问了该网站、什么时候进行的访问和访问的是什么内容等。



1. 访问日志的分类

在主配置文件/etc/apache2/apache2.conf 中已定义 5 个日志记录的格式：vhost_combined、combined、common、referer、agent，代码如下所示：

```
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
vhost_combined
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
```

也就是说，可将 Apache 的访问日志分为 5 类如表 18-2 所示。

表 18-2 访问日志的格式

格式分类	格式名称	说 明
虚拟主机综合日志格式	vhost_combined	在综合日志格式的基础上多保存了虚拟主机的主机名和端口号
综合日志格式	combined	结合以下三种日志信息
普通日志格式	common	常用的日志记录格式，大多数日志分析软件都支持这种格式
参考日志格式	referer	记录客户访问站点的用户身份
代理日志格式	agent	记录请求的用户代理

2. 日志格式说明符

在 LogFormat 指令中，出现了格式说明符，这些格式说明符的含义如表 18-3 所示。

表 18-3 LogFormat 格式说明符

格式说明符	说 明
%v	进行服务的服务器的标准名字 ServerName，通常用于虚拟主机的日志记录中
%p	虚拟主机的连接端口号
%h	客户机的 IP 地址
%l	从 Identd 服务器中获取远程登录名称，基本已废弃
%u	来自于认证的远程用户
%t	连接的日期和时间
%r	HTTP 请求的首行信息，典型格式是“METHODRESOURCEPROTOCOL”，即“方法资源协议”。经常可能出现的 METHOD 是 GET、POST 和 HEAD；RESOURCE 是指浏览器向服务器请求的文档或 URL；PROTOCOL 通常是 HTTP，后面再加上版本号，通常是 HTTP/1.1
%>s	响应请求的状态代码，一般这项的值是 200，表示服务器已经成功地响应浏览器的请求，一切正常；以 3 开头的状态代码表示由于各种不同的原因用户请求被重定向到了其他位置；以 4 开头的状态代码表示客户端存在某种错误；以 5 开头的状态代码表示服务器遇到了某个错误
%b	传送的字节数（不包含 HTTP 头信息），将日志记录中的这些值加起来就可以得知服务器在一天、一周或者一月内发送了多少数据
%{Referer}i	记录引用此资源的网页
%U	请求的 URL 路径，不包含查询串
%{User-Agent}i	使用的浏览器信息

补充说明

由于 LogFormat 指令中，格式字符串是包含在 “” 和 “” 之内的，或希望在记录信息时可使用双引号，则需要使用转义字符。如在 LogFormat 指令中设置了 \"%h\"，则在访问日志里的日志信息保存为 “"192.168.1.105"”。

3. 分析访问日志记录

默认的访问日志使用的是 combined 日志格式。则 /var/log/apache2/access.log 文件的内容如下所示：

```
....  
192.168.1.105 -- [30/Dec/2009:22:25:35 +0800] "GET / HTTP/1.1" 200 56 "-" "Mozilla/4.0  
(compatible; MSIE 7.0; Windows NT 5.1; GTB6.3; Mozilla/4.0 (compatible; MSIE 6.0; Windows  
NT 5.1; SV1); .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; CIBA; ODM)"  
....
```

此日志记录是按照 combined 格式说明符字符串生成的。combined 格式字符串定义如下：

```
%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"
```

以 combined 定义分析访问日志 access.conf 的内容，对应关系如表 18-4 所示。

表 18-4 说明符与内容的对应关系

格式说明符	记录数据
%h	192.168.1.105
%l	- (没有获取到信息)
%u	- (没有获取到信息)
%t	[30/Dec/2009:22:25:35 +0800]
%r	GET / HTTP/1.1
%>s	200
%b	56
%{Referer}i	- (没有获取到信息)
%{User-Agent}i	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; GTB6.3; Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1); .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; CIBA; ODM)

由上述信息可知客户端的 IP 地址是 192.168.1.105，请求连接的时间是 30/Dec/2009:22:25:35，代码 200 表示服务器已经成功地响应浏览器的请求，服务器传输了 56KB 的数据到客户端，客户端使用的浏览器是 Mozilla/4.0。

18.4.3 错误日志

错误日志和访问日志一样也是 Apache 的标准日志。错误日志是记录了服务器出错的细节和如何处理的信息。配置错误日志主要是通过 ErrorLog 和 LogLevel 指令来完成的。ErrorLog 指令配置错误日志保存的文件路径；LogLevel 指令配置错误日志的记录等级。错误日志等级如表 18-5 所示。



表 18-5 错误日志等级

紧急程序	等 级	说明
1	emerg	出现紧急情况使得系统不可用
2	alert	需立即引起注意的情况
3	crit	危险情况的警告
4	error	除了 emerg、alert、crit 的其他错误
5	warn	警告信息
6	notice	需引起注意的情况
7	info	值得报告的一般消息
8	debug	由运行于 debug 模式的程序所产生的信息

在主配置文件/etc/apache2/apache2.conf 中，可设置服务器中的错误日志；而在配置文件/etc/apache2/sites-available/default 中可设置针对每个虚拟主机的错误日志。默认起作用的错误日志文件为/var/log/apache/error.log，其保存了当前配置的虚拟主机所产生的错误信息记录。错误日志文件 error.log 的部分内容如下：

```
[Wed Dec 30 21:52:51 2009] [notice] caught SIGTERM, shutting down
[Wed Dec 30 21:53:56 2009] [notice] Apache/2.2.11 (Ubuntu) DAV/2 SVN/1.5.4 configured
-- resuming normal operations
[Wed Dec 30 21:54:00 2009] [error] [client 127.0.0.1] File does not exist: /htdocs
...
[Wed Dec 30 22:00:54 2009] [error] [client 192.168.1.105] File does not exist: /htdocs
[Wed Dec 30 22:01:20 2009] [error] [client 192.168.1.105] File d
```

一般来说，每条错误记录信息由 3 个部分组成：发生错误的时间、错误级别和错误信息主要内容。如上述最前的记录信息，[Wed Dec 30 21:52:51 2009] 表示发生错误的时间，[notice] 表示错误等级，caught SIGTERM,shutting down 是错误信息内容，表示捕获 SIGTERM 信息，关闭服务器。如果客户端连接到服务器上所产生的错误，则在错误记录信息上还会添加一段记录客户端的 IP 地址。如上述记录信息的后 3 行记录，添加了连接客户端的 IP 地址[client 192.168.1.105]等。

18.5 配置动态 Web 站点



Apache 服务器不仅能发布静态的站点，还可发布动态站点。在 Linux 环境下，常用的 Web 网站都可使用 Apache 进行发布。

PHP 是 PHP Hypertext Preprocessor 的缩写，是一种被广泛应用的开放源代码的多用途脚本语言，它可嵌入到 HTML 中，尤其适合 Web 开发。

在使用 PHP 时，需要安装 php5 软件包和 libapache-mod-php5 软件包。在终端输入如下代码：

```
root@dranyu-desktop:~# apt-get install php5 libapache2-mod-php5
```

php5 软件包为 Ubuntu 系统添加 php 脚本的解释器，而 libapache-mod-php5 则是为 Apache

服务器添加 php5 模块。安装完毕后，系统会在/etc/apache2/mods-available/目录添加 php5.load 和 php5.conf 两个配置文件。其中 php5.load 配置文件使 Apache 服务器加载了 php5 模块，其内容如下：

```
LoadModule php5_module /usr/lib/apache2/modules/libphp5.so //LoadModule 指令指定加载
php5 模块
```

加载 php5 的模块还需要对 php5 进行设置，设置文件 php5.conf 的内容如下：

```
<IfModule mod_php5.c> //定义模块的配置
    AddType application/x-httdp-php .php .phtml .php3 //添加文件类型，使得 Apache 能识别
    PHP 的文件
    AddType application/x-httdp-php-source .phps //添加.phps
</IfModule>
```

重启 Apache 服务器，在/var/www 目录下添加一个名为 index.php 的文件，内容如下：

```
<?php //php 的开始标签
phpinfo(); //Apache 的全局函数，用来显示 Apache 服务器和 PHP 的信息
?> //php 的结束标签
```

在浏览器地址栏上输入网址 http://localhost/index.php，若弹出如图 18-2 所示的页面，表示 Apache 已具备解释 PHP 的能力。

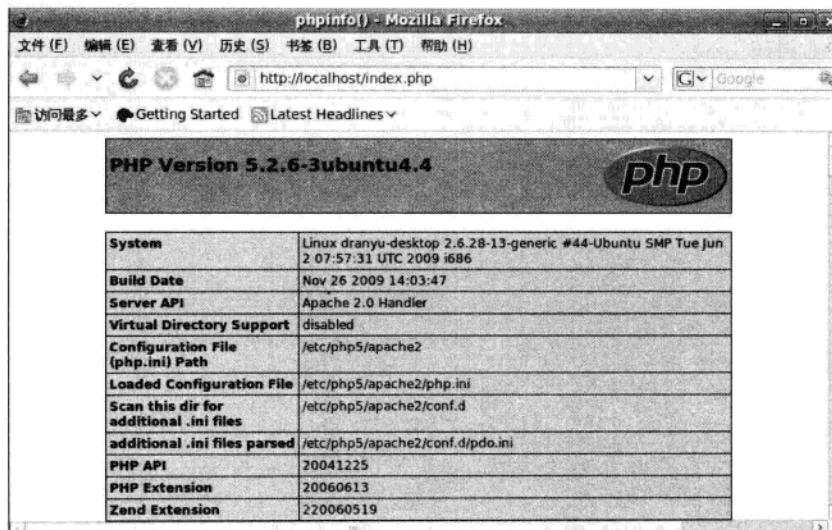
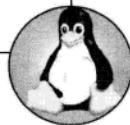


图 18-2 PHP 的信息界面

18.6 小结

本章简单介绍了两个流行的 Web 服务器：Apache 和 Tomcat。细说了 Apache 服务器的安装、配置和维护。详细解说了 Apache 的主要配置文件 apache2.conf 中出现的参数。还以例子的方式一步一步地教读者配置 Apache 服务器。



18.7 习题

1. 安装 Apache 软件包。
2. 使用 Gedit 打开配置文件/etc/apache2/apache2.conf，添加 ServerName 指令，将它的值设置为 127.0.0.1。
3. 试着将默认的监听端口号 80 修改为新的监听端口号 8088。
4. 修改保存网站资源的目录，将其修改到指定位置，如/home/dranyu/website。
5. 完成启动、停止、重启和查看运行状态等操作。
6. 在配置文件/etc/apache2/mods-available/alias.conf 中，为保存着一个网站资源的文件目录设置访问别名。
7. Apache 的标准日志有多少类？访问日志和错误日志的作用各是什么？
8. 访问日志的日志记录格式是由什么指令设置的？
9. 简述错误日志的错误等级。
10. 配置 PHP 环境。



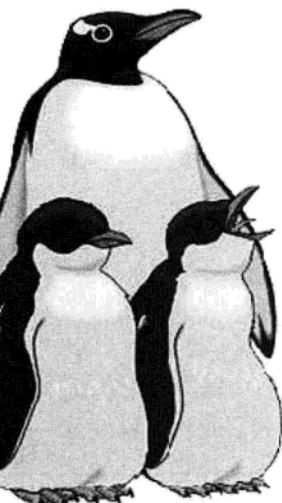
第19章

数据库 MySQL

Linux

数据库技术是信息技术中一个重要支撑，没有数据库技术，人们在浩瀚的信息世界中将显得手足无措。数据库最初是在大型企业作为大规模的事务处理的基础，后来随着个人计算机的发展，数据库技术被移植到 PC (Personal Computer) 上来，用于单用户个人数据处理。接着，由于网络技术的发展，PC 在工作组内联成网络，数据库技术就上升到工作组级。现在，数据库技术正被 Internet 和内联网的诸多应用使用着。数据库技术的地位不断在提升，那么对于一个系统管理员来说，掌握数据库的操作有利于系统的管理和维护。

本章讲述目前在中小企业应用广泛的 MySQL 数据库，主要介绍 MySQL 数据库在 Ubuntu 中的安装、配置和使用。





19.1 数据库简介

现在的数据库以关系数据库为主流，如 Microsoft SQL Server、Oracle、DB2、Mysql 和 PostgreSQL 等。Ubuntu 提供了 Mysql 和 PostgreSQL 的软件包。

19.1.1 MySQL 数据库

MySQL 是一个小型关系型数据库管理系统，开发者为瑞典 MySQL AB 公司。在 2008 年 1 月 16 日被 Sun 公司收购。目前，MySQL 被广泛地应用在 Internet 上的中小型网站中。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了 MySQL 作为网站数据库。MySQL 的官方网站的网址是：www.mysql.com。

MySQL 的特征如下所示：

- (1) 可移植性。它由 C 和 C++ 编写，并使用了多种编译器进行测试，保证 Mysql 源代码的可移植性。
- (2) 跨平台性。其支持 AIX、FreeBSD、HP-UX、Linux、Mac OS、Novell Netware、OpenBSD、OS/2 Wrap、Solaris、Windows 等多种操作系统。
- (3) 为多种编程语言提供了 API。这些编程语言包括 C、C++、Eiffel、Java、Perl、PHP、Python、Ruby 和 Tcl 等。
- (4) 支持多线程，采用了多 CPU 体系结构，充分利用 CPU 资源。
- (5) 优化的 SQL 查询算法，有效地提高查询速度。
- (6) 既能够作为一个单独的应用程序应用在客户端服务器网络环境中，又能够作为一个库而嵌入到其他的软件中提供多语言支持，常见的编码如中文的 GB 2312、BIG5，日文的 Shift_JIS 等都可以用作数据表名和数据列名。
- (7) 提供 TCP/IP、ODBC 和 JDBC 等多种数据库连接途径。
- (8) 高效管理数据，提供用于管理、检查、优化数据库操作的管理工具。
- (9) 多用户并发访问，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 仍然可保持着良好的性能。
- (10) 功能强大，可以处理拥有上千万条记录的大型数据库。

MySQL 数据库是典型的客户端/服务器结构系统，它由一个服务器守护程序 mysqld 和多种不同功能的客户端程序及函数库组成。

19.1.2 PostgreSQL 数据库

PostgreSQL 是一个功能强大的开源的关系数据库系统。它经过了 15 年多的开发历程，是一个经得起考验的体系结构，在可靠性、数据完整性和正确性方面赢得了极好的声誉。它运行在目前主要的操作系统上，包括 Linux、UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) 和 Windows。它是完全符合 ACID 原则，完全支持外键、联合、视图、触发器

和存储过程。它包含了 SQL92 和 SQL99 的大部分数据类型，包括 INTEGER、NUMERIC、BOOLEAN、CHAR、VARCHAR、DATE、INTERVAL 和 TIMESTAMP。同时，它也支持存储二进制巨型对象，包括图片、声音和视频。它拥有本地编程接口，支持与 C/C++、Java、.Net、Perl、Python、Ruby 和 Tcl 等编程的连接。

作为一个企业级类型的数据库，PostgreSQL 拥有先进的功能，例如多版本并发控制、时间点恢复、表空间、异步复制、嵌套事务、在线热备份、一个精密的查询计划/优化器、容错日志。它支持国际字符集、多字节字符编码、Unicode，它是本地感知的排序、大小写敏感和格式化。它是一个 TB 级的数据库，在有些生产环境中正在运行的 PostgreSQL 可管理超过 4TB 的数据。

19.2 构建 MySQL 数据库服务器



由于 MySQL 是开放的、免费的，所以几乎所有的 Linux 发行版本都集成了 MySQL 数据库的软件包。Ubuntu 自然也集成了 Mysql 数据库，在 Ubuntu9.04 版本上的 mysql 是 Mysql-5.0.75 版本，只不过 Ubuntu 系统在安装时并没有安装 Mysql 服务器。本节描述 MySQL 的安装、配置和使用。

19.2.1 安装 MySQL 数据库服务器

通过如下的命令可查看系统是否已经安装了 MySQL 数据库：

```
dranyu@dranyu-desktop:~$ dpkg -l | grep mysql
```

其中：dpkg -l 罗列系统已安装的所有软件包的记录；grep mysql 表示在系统安装的软件包记录中查找与 MySQL 匹配的软件包记录。若命令执行后，没有显示任何的记录表示 MySQL 数据库软件包并没有安装。要使用 MySQL 数据库，需要安装它，通常安装的方法有两种，第一种是在新立得软件包管理器中选择 mysql-server 软件包；第二种是在终端的命令提示符中输入如下命令：

```
root@dranyu-desktop:/home/dranyu# apt-get install mysql-server
```

执行命令结果如下所示：

```
root@dranyu-desktop:/home/dranyu# apt-get install mysql-server
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
将会安装下列额外的软件包： //注释：apt-get 命令自动添加安装 mysql-server 依赖的软件包
 libdbd-mysql-perl libdbi-perl libhtml-template-perl libnet-daemon-perl
 libplrpc-perl mysql-client-5.0 mysql-server-5.0 mysql-server-core-5.0
建议安装的软件包：
 dbishell libipc-sharedcache-perl mysql-doc-5.0 tinyca
下列【新】软件包将被安装：

您希望继续执行吗？[Y/n]y //注释：输入 y，确定安装 mysql 数据库
正在设置 mysql-server-5.0 (5.1.30really5.0.75-0ubuntu10.2) ...
 * Stopping MySQL database server mysqld [ OK ]
```



```
* Reloading AppArmor profiles ...
//注释: 启动mysql服务器的守护进程mysqld [ OK ]
* Starting MySQL database server mysqld [ OK ]
* Checking for corrupt, not cleanly closed and upgrade needing tables.
```

正在设置 libhtml-template-perl (2.9-1) ...
正在设置 mysql-server (5.1.30really5.0.75-0ubuntu10.2) ...

在安装 MySQL 服务器的过程中，安装程序要求系统管理员设置 MySQL 数据库 root 用户的密码，显示的对话框如图 19-1 所示。

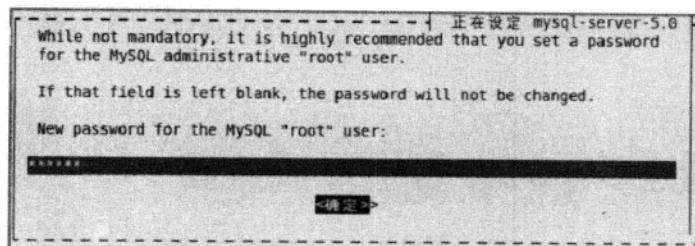


图 19-1 设定 MySQL 服务器 root 用户的密码

在对话框的输入栏中，输入 MySQL 服务器 root 的密码，然后按下 Tab 键，选中“确定”按钮，然后按下回车键，进入重复输入 root 的密码的界面，如图 19-2 所示。

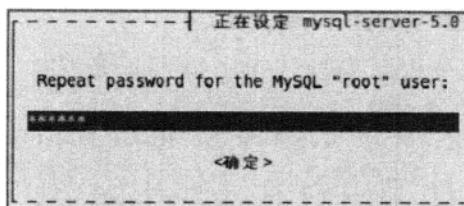


图 19-2 再次输入 root 的密码

按下 Tab 键，单击“确定”按钮，然后按下回车键，完成 root 用户密码的设置。

补充说明

在安装 MySQL 服务器软件包时，安装程序要求设置 root 用户的密码并不是 Ubuntu 系统中 root 用户的密码。要区分好 MySQL 服务器的 root 用户和 Ubuntu 系统的 root 用户。MySQL 服务器上的 root 用户拥有管理 MySQL 服务器中数据库的资源，如管理 MySQL 服务器的用户、添加删除服务器中的数据库等。Ubuntu 系统的 root 用户可管理 MySQL 服务器所占用的资源，设定哪个 Ubuntu 系统的用户可设置 MySQL 服务器，管理 MySQL 服务器的启动和关闭等相关问题。可以说 MySQL 管理员 root 用户和 Ubuntu 系统的管理员 root 是没有任何关系的。

19.2.2 配置 MySQL 数据库服务器

MySQL 服务器的使用如同其他的服务器软件包一样使用。

1. 使用 MySQL 用户和组账号

MySQL 服务器的软件包安装好之后，其会自动创建 MySQL 用户和 MySQL 组账号。这些用户和组账号是为了方便管理 MySQL 服务器上的文件和活动。这样在管理 MySQL 服务器的时候，系统管理员就不需要有 root 权限。查看 MySQL 用户和 MySQL 组账号的信息如下：

```
root@dranyu-desktop:/var/log# cat /etc/passwd | grep mysql //显示 MySQL 用户账号的信息
mysql:x:117:MySQL Server,,,:/var/lib/mysql:/bin/false
root@dranyu-desktop:/var/log# cat /etc/group | grep 127 //显示 MySQL 组的信息
mysql:x:127:
```

从/etc/passwd 文件的内容中，可知道 MySQL 的 UID 是 117，其 GID 是 127。“MySQL Server”字符串是 MySQL 用户的标识。MySQL 的主目录指定为/var/lib/mysql，此目录保存 MySQL 服务器定义的每个数据库信息。MySQL 用户的默认 shell 是/bin/false。

MySQL 组账号的信息保存在/etc/group 文件中，显示了组名为 MySQL 和其的 GID 为 127。

MySQL 用户账号和 MySQL 组账号是 Ubuntu 系统的账号，对 MySQL 服务器进行维护和管理。若要分配给 Ubuntu 系统的用户管理 MySQL 服务器的权限，就需要使 Ubuntu 系统用户是 MySQL 组里的一员，将 Ubuntu 用户添加到 MySQL 组中的方法如例 19-1 所示。

例 19-1：将 Ubuntu 用户 dranyu 添加到 MySQL 组

root@dranyu-desktop:/var/lib/mysql# usermod -G mysql dranyu //注释：使用 usermod 命令修改

用户 dranyu 的属性，-G 表示修改 dranyu 的所从属的其他组

命令执行后，dranyu 用户就会被添加到/etc/group 文件中，如下所示：

```
root@dranyu-desktop:/var/lib/mysql# cat /etc/group | grep mysql
mysql:x:127:dranyu //表示 dranyu 是 MySQL 组的一个成员
```

这样，dranyu 就拥有了管理 MySQL 服务器的权限了。

2. 修改 MySQL 管理员的密码

在安装时，MySQL 管理员 root 的密码就已被设置了。若用户需要修改 MySQL 管理员 root 的密码，需要使用如下格式的 mysqladmin 命令。

```
mysqladmin -u root -p password 新密码
```

其中-u 是 mysqladmin 的选项，指定登录 MySQL 服务器的用户名为 root；-p 选项表示 root 登录是需要输入密码进行认证；Password 命令表示设置密码，而“新密码”表示 MySQL 管理员 root 的新密码。

例 19-2：将 MySQL 管理员 root 用户密码修改为“sk1234”

修改操作所执行的命令如下：

```
root@dranyu-desktop:/var/lib/mysql# mysqladmin -u root -p password sk1234
Enter password: //注释：输入 MySQL 管理员 root 的旧密码，按下回车键
```

设置好了 MySQL 管理员密码，可保证 MySQL 数据库中数据的安全，因为 MySQL 管理员 root 对 MySQL 服务器中的数据库拥有所有的控制权限。

3. MySQL 服务器的主配置文件

MySQL 服务器的主配置文件为/etc/mysql/my.cnf，其设置了 MySQL 服务器运行时使用的全局变量的值。使用 MySQL 服务器，设置好主配置文件是十分重要的。

/etc/mysql/my.cnf 配置文件由节组成，每个节指定了由 MySQL 服务器中特定的进程读取。节的编写格式如下：

[节名]



属性 1=属性值 1

.....

属性 n=属性值 n

例 19-3：client 节的编写

client 节的属性会被所有的 MySQL 客户端在运行时所读取，其在/etc/mysql/my.cnf 配置文件中的默认内容如下：

```
[client]
port      = 3306
socket    = /var/run/mysqld/mysqld.sock
```

读取 client 节属性的进程包括 mysql、mysqladmin、mysqldump、mysqlimport 和 mysqlshow 等。“[client]”说明此节为 client 节，并标识着上一个的结束同时标识着此节的开始。“port=3306”表示 port 属性的值是 3306，也就是指定了 MySQL 客户端接收数据的端口号为 3306。“socket=/var/run/mysqld/mysqld.sock”设定 MySQL 客户程序与服务器之间的本地通信的一个套接字文件，socket 是属性名，/var/run/mysqld/mysqld.sock 是套接字文件的绝对路径。

默认的/etc/mysql/my.cnf 配置文件配置的是一个小型的 MySQL 服务器，它的内容如下：

```
#                                         # The MySQL database server configuration file.
#
#                                         [client]          //client 节
port      = 3306
socket    = /var/run/mysqld/mysqld.sock
#
#                                         [mysqld_safe]      //mysqld_safe 节，相应的进程为 mysqld_safe
socket    = /var/run/mysqld/mysqld.sock //设置套接字文件
nice     = 0   //设置使用 nice 程序来为服务器的运行级别指定一个数值，此处设置为 0
#
#                                         [mysqld]          //MySQL 服务器守护进程 MySQL 所读取的 MySQL 节，设置整个服务器的数据
保存与 MySQL 服务器的使用状况
#
socket    = /var/run/mysqld/mysqld.sock //设置套接字文件的路径
port      = 3306 //MySQL 服务器的端口号
basedir   = /usr //使用给定目录/usr 作为根目录，表示 MySQL 安装在/usr 目录下
datadir   = /var/lib/mysql //从给定目录/var/lib/mysql 读取数据库，并保存数据库
bind-address = 127.0.0.1 // MySQL 服务器的 IP 地址。如果 MySQL 服务器所在的
计算机有多个 IP 地址，这个选项将非常重要
key_buffer      = 16M           //指定用来存放索引区块的 RMA 值，默认值为 16M
max_allowed_packet = 16M        //设置客户与服务器之间交换的数据包的最大长度，这个数字至
少应该大于客户程序将要处理的最大 BLOB 块的长度，默认为
16M
thread_stack      = 128K         //设置线程堆栈的大小，默认为 128K
thread_cache_size = 8            //设置线程缓存的大小
query_cache_limit  = 1M           //设置允许临时存放在查询缓存区里的查询结果的最大长度
query_cache_size   = 16M          //设置查询缓存区的最大长度
...
```

4. 主配置文件的示范文件

MySQL 主配置文件是/etc/mysql/my.cnf，是当前 MySQL 服务器使用的配置文件，而 MySQL 应用程序还为 MySQL 服务器预留了主配置文件的示范文件，也就是说使用示范的配置文件代替/etc/mysql/my.cnf 主配置文件就可配置为示范配置文件中定义的 MySQL 服务器。

这些示范文件保存的位置如下所示：

```
root@dranyu-desktop:/usr/share/doc/mysql-server-5.0/examples# ls -l //罗列详细的文件
信息
总用量 28
-rw-r--r-- 1 root root 2104 2009-05-14 18:39 my-huge.cnf.gz
-rw-r--r-- 1 root root 8148 2009-05-14 18:39 my-innodb-heavy-4G.cnf.gz
-rw-r--r-- 1 root root 2090 2009-05-14 18:39 my-large.cnf.gz
-rw-r--r-- 1 root root 2118 2009-05-14 18:39 my-medium.cnf.gz
-rw-r--r-- 1 root root 2529 2009-05-14 18:39 my-small.cnf
-rw-r--r-- 1 root root 723 2009-05-14 18:39 ndb_mgmd.cnf
```

在显示结果中，以“.gz”为后缀的文件是使用 gzip 命令压缩的文件，使用如下格式的命令就可将指定的文件解压：

```
gzip -d 压缩文件名
```

例 19-4：解压 my-large.cnf.gz 文件的命令如下：

```
root@dranyu-desktop:/usr/share/doc/mysql-server-5.0/examples# gzip -d
my-large.cnf.gz
```

解压后的文件名为 my-large.cnf，就是一个文本文件，作为 MySQL 服务器主配置文件的示范文件。

各示范文件的功能如下：

- my-small.cnf

此示范配置文件是为了小型数据库而设计的。它应用于内存小于 64M 的系统，使用此配置后，MySQL 的守护进程不能占用太多的系统资源，也就限制了 MySQL 的部分功能。

- my-medium.cnf

此示范配置文件是为中等规模的数据库而设计的。使用此配置文件后，若系统的内存有 32M~64M 的范围，那么此主机只能安装 MySQL 一个服务器；如果系统内存有 128M 以上，系统除了可使用 MySQL 服务器，还可运行其他服务器，如 Web 服务器。

- my-large.cnf

此示范配置文件是为专用于一个 SQL 数据库的计算机而设计的。由于它可以为该数据库使用多达 512MB 的内存，所以在这种类型的系统上将需要至少 1GB 的内存，以便它能够同时处理操作系统与数据库应用程序。此配置文件对系统的硬件要求比较高。

- my-huge.cnf

此示范配置文件是为企业中的数据库而设计的。这样的数据库要求使用专用服务器和 1GB 或 1GB 以上的内存。

- my-innodb-heavy-4G.cnf

此示范配置文件要求系统的内存有 4G 或 4G 以上。将 MySQL 服务器配置为使用 Innodb 存储引擎以实现执行复杂的数据查询。

- ndb_mgmd.cnf

此示范配置文件将 MySQL 服务器配置为 MySQL 集群服务器，此配置是为企业适合大规模应用而设计的。

这些选择高度依赖于内存的数量、计算机的运算速度、数据库的细节大小、访问数据库的用户数量，以及在数据库中装入并访问数据的用户数量。系统管理员在使用这些示范配置



时，也就需要考虑到上述的依赖条件。

例 19-5：使用 my-small.cnf 示范配置文件代替当前主配置文件

要使示范配置文件起作用，需要将相应的示范配置文件作为 MySQL 服务器的主配置文件，以使用 my-small.cnf 为例，修改 MySQL 的主配置文件，具体的操作步骤为：

- (1) 保存原有的主配置文件/etc/mysql/my.cnf，将其备份为 my.cnf.old，执行如下命令：

```
root@dranyu-desktop:/etc/mysql# cp my.cnf my.cnf.old
```

- (2) 使用 cp 命令将示范配置文件 my-small.cnf 复制到/etc/mysql 目录下，同时重命名为 my.cnf，将原来的 my.cnf 主配置文件覆盖，执行如下命令：

```
root@dranyu-desktop:/etc/mysql# cp /usr/share/doc/mysql-server-5.0/examples/my-small.cnf my.cnf
```

- (3) 示范配置文件代替主配置文件之后，需要重启系统或者重启 MySQL 应用程序。如此就将示范配置文件 my-small.cnf 应用到了 MySQL 服务器上。

19.2.3 启动和关闭 MySQL 数据库服务器

在 Ubuntu 中，MySQL 服务器在安装完成后就已启动。在修改配置文件后，需要重启 MySQL 服务器，所设置的配置才会起作用。此时，需要使用到 MySQL 应用程序提供的 /etc/init.d/mysql 启动脚本。MySQL 启动脚本的基本语法如下：

```
/etc/init.d/mysql [选项]
```

使用的选项包括：start、stop、restart、reload、force-reload、status。以下举例说明常用的选项的含义。

例 19-6：启动 MySQL 服务器

```
root@dranyu-desktop:/etc/mysql# /etc/init.d/mysql start //start 选项表示启动 MySQL 服务器
* Starting MySQL database server mysqld [ OK ]
```

例 19-7：停止 MySQL 服务器

```
root@dranyu-desktop:/etc/mysql# /etc/init.d/mysql stop //stop 选项表示停止 MySQL 服务器
* Stopping MySQL database server mysqld [ OK ]
```

例 19-8：重启 MySQL 服务器

```
root@dranyu-desktop:/etc/mysql# /etc/init.d/mysql restart //restart 选项表示重启 MySQL 服务器
* Stopping MySQL database server mysqld [ OK ]
* Starting MySQL database server mysqld [ OK ]
* Checking for corrupt, not cleanly closed and upgrade needing tables.
```

例 19-9：查看 MySQL 服务器的运行状态

```
root@dranyu-desktop:/etc/mysql# /etc/init.d/mysql status //status 选项表示查看 MySQL 服务器的运行状态
* /usr/bin/mysqladmin Ver 8.41 Distrib 5.0.75, for debian-linux-gnu on i486
Copyright (C) 2000-2006 MySQL AB
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL license

Server version      5.0.75-0ubuntu10.2    //显示 MySQL 服务器所使用的 MySQL 版本
Protocol version  10
Connection          Localhost via UNIX socket //使用的套接字的类型
UNIX socket        /var/run/mysqld/mysqld.sock //套接字文件的路径
```

```

Uptime:          1 min 14 sec      //更新的时间

Threads: 1  Questions: 78  Slow queries: 0  Opens: 98  Flush tables: 1  Open tables: 4
Queries per second avg: 1.054

```

19.3 使用 MySQL 数据库



MySQL 应用程序是一个符合 SQL-92 或 SQL-2 标准的关系型管理系统，其使用 SQL 结构化查询语言进行数据库管理、用户管理等操作。MySQL 数据库的管理操作都使用了 SQL，并且是在 MySQL 命令提示符“mysql>”下进行的，每条语句以“;”或“\g”结束，而且输入不区分大小写，如 Use 与 use 表示同一个意义。

19.3.1 运行 MySQL 命令

在 Ubuntu 的终端上 mysql 数据库，那么 MySQL 命令就是入口。在任何终端的命令提示符上输入如下格式的命令就可进入到 MySQL 服务器的字符操作界面：

```
mysql -u mysql 用户名 -p
```

其中-u 选项指定登录的用户为 MySQL 用户名；-p 选项表示使用密码登录。

例 19-10：以 root 身份登录 MySQL 服务器的字符操作界面的命令如下：

```
root@dranyu-desktop:/usr/share/doc/mysql-server-5.0/examples# mysql -u root -p //输入命令后按下回车键
```

```
Enter password: //输入 root 用户的密码，然后按下回车键
```

```
//显示使用说明，以“;”或“\g”结束输入的命令
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Your MySQL connection id is 33 //显示当前的连接 ID 为 33
```

```
Server version: 5.0.75-0ubuntu10.2 (Ubuntu) //显示 MySQL 服务器应用程序的版本
```

```
Type 'help;' or 'h' for help. Type '\c' to clear the buffer. //表示键入 help 或\h 获取帮助，键入\c
```

```
清空缓冲区的数据
```

```
mysql> //MySQL1 字符操作界面的命令提示符，在输入相应的命令，然后按下回车键
```

即可，MySQL 应用程序就会解析并执行相应的命令

在使用 MySQL 数据库之前，先检查 MySQL 的运行状态，可在 MySQL 的命令提示符后输入 status 命令显示当前运行状态信息如下所示：

```

mysql> status;
-----
mysql Ver 14.12 Distrib 5.0.75, for debian-linux-gnu (i486) using readline 5.2

Connection id:      33 //当前的连接 ID
Current database: 
Current user:      root@localhost //当前登录的用户，root 用户在本地主机上登录，localhost 表示主机的名称
SSL:               Not in use //没有使用 SSL 认证
Server version:    5.0.75-0ubuntu10.2 (Ubuntu) //服务器应用程序的版本号为 5.0.75
Protocol version: 10 //协议的版本号为 10
Connection:        Localhost via UNIX socket //套接字的类型

```



```
//characterset 表示字符集
Server characterset: latin1
Db      characterset: latin1
Client characterset: latin1
Conn. characterset: latin1
UNIX socket:        /var/run/mysqld/mysqld.sock //套接字文件的路径
Uptime:            1 hour 16 min 59 sec //服务器运行的时间

Threads: 1 Questions: 95 Slow queries: 0 Opens: 98 Flush tables: 1 Open tables: 4
Queries per second avg: 0.021
-----
```

状态信息显示了 MySQL 服务器的版本号为 5.0.75，还显示一当前用户 root@localhost 正在操作的数据库，此处为空表示 root 用户还没选择操作的数据库。

补充说明

使用 MySQL 时的一些术语，如 threads 表示当前运行的进程数；query 表示 MySQL 的查询记录；tables 表示数据库的数据表；database 表示服务器上的数据库；Questions 表示用户向服务器运行了多少次查询。

19.3.2 数据库的管理

MySQL 服务器上可保存多个数据库，合理地管理数据库必然要使用到数据库管理的命令。

1. 显示服务器上所有的数据库

MySQL 服务器管理着多个数据库，在使用指定的数据库之前，需要知道 MySQL 服务器是否保存了用户所需要的数据库或有时系统管理员不记得一个特定数据库的名称，这时可使用“show databases”；命令显示 MySQL 服务器上的数据库。

例 19-11：显示当前服务器上保存的所有数据库

```
mysql> show databases; //show 命令，功能：显示当前服务器上的所有数据库
+-----+
| Database      | //注释：Database 是查询结果列表的表头
+-----+
| information_schema | //第一条记录，表示 MySQL 服务器中有 information_schema 数据库
| mysql          | //第二条记录，表示拥有 MySQL 数据库
+-----+
2 rows in set (0.00 sec) //查询结果显示 2 条记录，发了 0.00 秒的查询时间
```

这一显示所有的命令，可帮助系统管理员方便了解此服务器上的所有数据库。

补充说明

information_schema 数据库和 MySQL 数据库是 MySQL 的两个默认添加的数据库。information_schema 数据库是信息数据库，其中保存着关于 MySQL 服务器所维护的所有其他数据库的信息，其提供了访问数据库元数据的方式。MySQL 数据库保存了 MySQL 服务器的用户信息，保存了 MySQL 服务器使用到的相关数据。元数据是关于数据的数据，如数据库名或表名，列的数据类型，或访问权限等。

2. 选择数据库

使用 show 命令罗列了服务器上的所有数据库，若要使所有的事务应用到指定的数据库，总需要选择指定的数据库进行操作，在 MySQL 服务器中选择或切换当前数据库可使用命令格式如下所示：

```
use 数据库名称;
```

其中数据库名称是当前服务器上保存的数据库名称，即使用 show 命令显示列表中数据库的名称。

例 19-12：选择 MySQL 数据库

MySQL 数据库的数据库名称在 show 命令的结果显示列表中，选择 MySQL 数据库的命令如下所示：

```
mysql> use mysql; //选择 MySQL 数据库
Database changed
mysql> status; //显示当前连接的状态信息
-----
...
Current database: mysql //当前使用的数据库为 MySQL 数据库
Current user: root@localhost
...
-----
```

进入 MySQL 数据库后，就可在 MySQL 数据库下执行其他的事务，如显示此数据库中数据表、创建数据表、插入数据记录、删除数据记录和查询数据记录。

3. 创建数据库

在 MySQL 服务器默认的数据库是用来管理 MySQL 服务器的，其他应用程序需要使用 MySQL 服务器保存数据时，系统管理员需要为应用程序创建相应的数据库。在 MySQL 服务器中创建数据库语句的基本格式为：

```
create database 新数据库的名称;
```

其中 create 是 SQL 的一个创建命令，database 表示所创建的对象是一个数据库，新数据库的名称指定此新创建的数据库的名称。在 Linux 下数据库的名称是区分大小写的，而在 MySQL 服务器上，数据库名称不可重名。

例 19-13：创建一个名为 web3ds 的 web3ds 网站数据库，可输入如下命令：

```
mysql> create database web3ds; //输入命令，按下回车键
Query OK, 1 row affected (0.00 sec)
```

查看 web3ds 数据库是否已被创建，使用如下命令：

```
mysql> show databases; //显示所有的数据库
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| web3ds         | //新创建的数据库 web3ds
+-----+
3 rows in set (0.00 sec)
```

数据库 web3ds 创建之后，数据库 web3ds 的相应数据文件保存在 /var/lib/mysql 目录下的 web3ds 目录里。使用 ls 命令显示 /var/lib/mysql 的内容如下：

```
root@dranyu-desktop:/var/lib/mysql# ls -l
```



```
总用量 20496
```

```
drwxr-xr-x 2 mysql root      4096 2009-12-01 10:52 mysql
drwx----- 2 mysql mysql    4096 2009-12-05 12:19 web3ds
```

结果显示列表中，文件夹 web3ds 就与 web3ds 数据库相对应。

补充说明

MySQL 服务器上每个应用程序的数据库文件都单独地保存在 /var/lib/mysql 目录的相应数据库名称的目录里。如 web3ds 数据库的数据库文件保存在 /var/lib/mysql 目录下的 web3ds 目录里。这样方便数据库的移植。数据库移植时，只需将 /var/lib/mysql 下相应的数据库名目录复制到其他 MySQL 服务器的数据库保存目录（/var/lib/mysql）下，在其他 MySQL 服务器上就可重用此数据库。如一个使用 MySQL 作为数据库系统的网站，将此网站从测试机上布置到生产机上，将保存在测试机的数据库直接复制到生产机的 MySQL 服务器上就完成网站数据库的部署。

4. 删除数据库

为了可清除不使用的数据库，系统管理需要删除服务器上不必要的数据库。在删除一个数据库的同时会将其管理的数据表也删除，所以在执行数据库的删除操作时需要特别小心。删除一个数据库命令的基本语法格式为：

```
drop database 数据库名称;
```

其中 drop 是 MySQL 中的一个删除命令； database 表示所针对的操作对象是数据库；数据库名称指的是已存在服务器上的数据库名称。

例 19-14：删除例 19-13 所创建的数据库 web3ds，命令如下所示：

```
mysql> drop database web3ds; //输入命令，按下回车键
Query OK, 0 rows affected (0.01 sec)
```

查看 web3ds 数据库是否已被删除，使用如下命令：

```
mysql> show databases; //显示 MySQL 服务器上所有的数据库
+-----+
| Database      |
+-----+
| information_schema |
| mysql          | //只有两个默认的数据库，web3ds 数据库已被删除
+-----+
2 rows in set (0.00 sec)
```

19.3.3 表的管理

在关系型数据库中，数据库是由多个数据表组织起来的，这些数据表用来存储数据。数据表是一系列二维数组的集合，用来代表和储存数据对象之间的关系。它由纵向的列和横向的行组成，例如，一个班级的学生名单的信息表，每一列都包括了所有学生的一个特定信息，如学号；而每一行则包括了一个学生的所有信息，如学号、姓名、成绩等。

对于特定的数据表，列的数目一般事先固定，各列之间可以由列名来识别。而行的数目可以随时、动态变化，每行通常都可以根据某个（或某几个）列中的数据来识别，称为候选键。没有数据表的数据库是不用的，添加了表，才能有组织、有规律地管理数据和存储数据。

1. 创建数据表

在使用 `create` 命令创建新的数据库之后，数据库是空的。没有任何的数据，也不能保存数据。选择数据库后，使用如下的命令格式可显示当前数据库中保存的数据表：

```
show tables;
```

例 19-15：显示当前数据库中所有数据表的信息，命令执行和结果显示如下所示：

```
mysql> use web3ds; //选择 web3ds 数据库
Database changed
mysql> show tables; //显示 web3ds 数据库中的数据表信息
Empty set (0.00 sec)
“Empty set (0.00sec)” 表示此时 web3ds 数据库中没有任何的数据表结构
```

在 MySQL 字符操作界面中，可使用以下格式的语句创建数据表：

```
create table 表名 (字段 1, 字段 2.....字段 n, [表级约束]) [ ENGINE=引擎类型]
```

`create table` 命令的很多其中常用的选项与参数含义如下：

- `create table` 是创建新数据表的 SQL 语句。
- 表名设定所要创建的数据表名称。
- 字段 1~字段 n 指定数据表的列名，并且定义了字段的数据类型与约束，格式如下：
字段名 数据类型 [字段修饰符]
- 表级约束指定表的主键、外键、索引和唯一约束等。
- 引擎类型用于指定表中数据的存储格式。

MySQL 数据库中规定了一些字段的数据类型如 Int 整数类型等，其中常用的数据类型如表 19-1 所示。对字段的修饰符，MySQL 数据库也做了规定，其中常用的字段约束如表 19-2 所示。表级约束影响整个表的存储，表级约束常用的选项如表 19-3 所示。

表 19-1 MySQL 数据库的常用数据类型

数据类型	描述
CHAR (M)	用于表示 M 长度的字符串，所支持的最大长度是 255 个字符
VARCHAR(M)	用于保存可变长度的字符串，VARCHAR 数据类型所支持的最大长度也是 255 个字符
INT (M) [unsigned]	INT 数据类型用于保存从 -2147483647 到 2147483648 范围之内的任意整数数据。如果用户使用 Unsigned 选项，则有效数据范围调整为 0-4294967295
FLOAT [(M,D)]	FLOAT 数据类型用于表示数值较小的浮点数据，可以提供更加准确的数据精度。其中，M 代表浮点数据的长度（即小数点左右数据长度的总和），D 表示浮点数据位于小数点右边的数值位数
DATE	此数据类型用于保存日期数据，默认格式为 YYYY-MM-DD
TEXT / BLOB	TEXT 和 BLOB 数据类型可以用来保存 255 到 65535 个字符，TEXT 不区分大小写，而 BLOB 对字符的大小写敏感，并且 BLOB 保存可变的数据的二进制类型

表 19-2 常用的字段修约束

约束名称	描述
Null (或 Not Null)	指定该字段的值为空(NULL)，或者必须填上数据(NOT NULL)
DEFAULT	指定该字段的默认值
AUTO_INCREMENT	只适用于 INT 字段，表明 MySQL 应该自动为该字段生成一个数，每次在上一次生成的数值上面加 1。在一张表中，只允许一个字段拥有此修饰符



表 19-3 表级约束

约束名称	描述
PRIMARY KEY	指定表的主键
FOREIGN KEY ...REFERENCES ...	指定表的外键约束
INDEX	为表创建一个索引
UNIQUE	指定索引的字段。被建立索引的字段，其值必须唯一

MySQL 支持多种存储引擎作为对不同表的类型的处理器。MySQL 存储引擎包括处理事务安全表的引擎和处理非事务安全表的引擎，常用的存储引擎如下：

- MyISAM

此存储引擎管理非事务表。它提供高速存储和检索，以及全文搜索能力。MyISAM 在所有 MySQL 配置里被支持，它是默认的存储引擎，除非你配置 MySQL 默认使用另外一个引擎。

- MEMORY

此存储引擎提供“内存中”表。MERGE 存储引擎允许集合将被处理同样的 MyISAM 表作为一个单独的表。就像 MyISAM 一样，MEMORY 和 MERGE 存储引擎处理非事务表，这两个引擎也都被默认包含在 MySQL 中。

- InnoDB 和 BDB

这两个存储引擎提供事务安全表。BDB 被包含在为支持它的操作系统发布的 MySQL-Max 二进制分里。

- EXAMPLE

此存储引擎是一个“存根”引擎，它不做任何事情。你可以用这个引擎创建表，但没有数据被存储于其中或从其中检索。这个引擎的目的是服务，在 MySQL 源代码中的一个例子，它演示说明如何开始编写新存储引擎。同样，它的主要兴趣是对于开发者。

例 19-16：在数据库 web3ds 中，创建一个 userInfo 的表用来保存登录到网站的用户信息，命令如下所示：

```
mysql> create table userInfo( //指定新创建的表名为 userInfo
    -> userId char(18) not null, //定义字段 userId, 它的数据类型为 char, 长度为 18,
并且其值不能为空
    -> userName varchar(20) not null, //定义字段 userName, 数据类型为 varchar, 长度为 20,
并且它的值不能为空
    -> sex char(1) default 'b', //定义字段 sex, 数据类型为 char, 长度为 1, 设置默认值为 b
    -> birthday date, //定义字段 birthday, 数据类型为 date 日期类型
    -> primary key (userId) //设置 userId 为 userInfo 表的主键
    -> )ENGINE=MYISAM; //指定存储引擎为 MYISAM, 输入 ";" 结束命令的输入
Query OK, 0 rows affected (0.01 sec)
```

在输入创建数据表的命令时，由于命令比较长，为了输入可在输入命令时，在适当的位置按下回车键将命令分为多行输入，最后还是以“;”符号结束命令的输入。创建完成后，使用 show 命令显示数据库 web3ds 上所有数据表，如下所示：

```
mysql> show tables;
+-----+
| Tables_in_web3ds | //这是结果列表的表头, 意识是 web3ds 数据库中的表
+-----+
```

```
| userInfo      | //新添加的 userInfo 表
+-----+
1 row in set (0.00 sec)
```

数据表创建后，系统管理员还使用 `describe` 命令查看数据表的结构，检查数据表的结构是否正确。`describe` 命令的使用方式如下：

```
describe 表名;
```

例 19-17：显示 `userInfo` 表的结构，执行命令和显示结果如下所示：

```
mysql> describe userInfo;
+-----+-----+-----+-----+-----+
| Field   | Type    | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| userId  | char(18) | NO   | PRI | NULL    |       |
| userName | varchar(20) | NO   |     | NULL    |       |
| sex     | char(1)   | YES  |     | b       |       |
| birthday | date    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

显示结果列表中，`Field` 表示字段的名称，`Type` 表示数据类型，`Null` 表示值是否可为空，`Key` 表示字段是否为主键或外键，`default` 表示是否设置了默认值，`Extra` 表示对 `userInfo` 表中每个字段的备注。

补充说明

当数据表中的字段设置不为空而且没有设置默认值时，系统会为此字段设置一个默认值为 `NULL`。`userInfo` 数据表使用了 MYISAM 存储引擎，那么在创建了此表之后，MySQL 应用程序会在 `/var/lib/mysql/web3ds` 文件目录下创建 3 个文件：`userInfo.frm` 表定义文件、`userInfo.MDY` 数据文件和 `userInfo.MYI` 索引文件。

2. 复制数据表

在 MySQL5.0 中，MySQL 服务器提供了完整复制表的语句，复制的同时也复制了表的结构。此语句的格式为：

```
create table 新表名 like 源表名;
```

例 19-18：将 `userinfo` 表复制为另一个 `vip` 表，可使用如下命令：

```
mysql> create table vip like userInfo; //输入命令，按下回车键，执行命令
Query OK, 0 rows affected (0.00 sec)
```

完成操作后，显示当前 `web3ds` 数据库中所有表信息如下所示：

```
mysql> show tables;
+-----+
| Tables_in_web3ds |
+-----+
| userInfo          |
| vip              | //新建的 vip 表
+-----+
2 rows in set (0.00 sec)
```

使用复制表功能就可快速地创建相似的数据表，在复制了的 `vip` 表上修改一些字段即可成为新的数据表。



3. 删除数据表

MySQL 数据库管理系统中，使用 drop 命令可删除一个或多个数据表。drop 命令删除数据表的格式为：

```
drop table 表名称 1[, 表名称 2];
```

例 19-19：删除表 vip，可使用下面的命令。

```
mysql> drop table vip;  
Query OK, 0 rows affected (0.00 sec)
```

完成删除表操作，显示 web3ds 数据库上所有表信息，显示结果如下：

```
mysql> show tables;  
+-----+  
| Tables_in_web3ds |  
+-----+  
| userInfo |  
+-----+  
1 row in set (0.00 sec)
```

注意

删除命令是不可恢复的命令。在删除数据表之前一定要考虑清楚。使用 drop table 删除数据表，删除数据表的同时删除表中的所有数据记录。数据库构建初期可删除数据表，若数据库已投入生产一般不会删除其中的表。因为投入生产后表中已有相关的历史数据，企业会保存相关的历史数据。

4. 修改数据表

在创建表之后，如果要修改表结构，如添加、删除或者修改表字段，创建或撤销索引，更改表的名称和类型等，则需要使用 alter 语句来进行。alter 语句的基本格式为：

```
alter table 表名 更改动作 1[, 更改动作 2, ...];
```

这里的更改动作是由 add、drop、change、alter、modify 和 rename 等关键字以及有关字段的定义组成。下面通过实例来介绍 alter 命令的一些具体使用方法。

例 19-20：为数据表增加新的字段

为例 19-18 由 userInfo 所复制出来的 vip 增加一个字段 address，将 address 字段设置为 varchar 数据类型，长度为 25，需要执行的命令如下：

```
mysql> alter table vip add address varchar(25); //alter table 是命令，vip 为表名，add 是  
更改动作的关键字表示添加元数据。address 表示新字段  
的字段名，varchar(25) 表示数据类型是 varchar。  
25 为长度  
Query OK, 0 rows affected (0.04 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
mysql> describe vip; //显示 vip 表的结构  
+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+  
| address | varchar(25) | YES | | NULL | //新增加的 address 字段  
+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

例 19-21：更改数据表的字段名和字段类型

要将 vip 数据表中字段 address 的名称改为 remark，并将该字段的数据类型改为 text，可使用下面的命令，同时显示数据表结构：

```
mysql> alter table vip change address remark text; //change 关键字表示修改指定的数据表，如 vip, address 是 vip 已有的字段, remark 是修改后的字段名称, text 是此字段的数据类型
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe vip; //显示 vip 表结构
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| remark | text      | YES  |     | NULL    |           | //address 字段已被删除, 修改为
了 remark, 数据类型为 text。
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

注意

即使仅仅改变字段名而不改变字段类型，change 子句后面也必须给出该字段的字段类型。

例 19-22：只修改字段的类型

要将表 vip 中字段 remark 的类型由 text 改为 varchar，命令执行和结果显示如下所示：

```
mysql> alter table vip modify remark varchar(25); //modiy 关键字修改指定字段 remark 的
数据类型, 将数据类型修改为 varchar(25)
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> describe vip; //显示 vip 数据表的结构
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| remark | varchar(25) | YES  |     | NULL    |           | //数据类型已修改为 varchar(25)
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

例 19-23：删除数据表的字段

使用 drop 关键字，删除 vip 数据表中的 remark 字段，命令执行和结果显示如下所示：

```
mysql> alter table vip drop remark; //drop 是关键字, 指定的字段为 remark
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> describe vip; //显示 vip 数据表的结构
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec) //remark 字段已被删除
```

例 19-24：重命名数据表

使用 rename 关键字可重命名数据表，将 vip 数据表重命名为 vipInfo，命令执行和结果显示如下所示：



```
mysql> alter table vip rename vipInfo; //rename 是关键字, vipInfo 是 vip 数据表的新名称
Query OK, 0 rows affected (0.00 sec)
mysql> show tables; //显示 web3ds 数据库中所有数据表
+-----+
| Tables_in_web3ds |
+-----+
| userInfo          |
| vipInfo           | //已改名为 vipinfo 的数据表
+-----+
2 rows in set (0.00 sec)
```

19.3.4 数据记录的管理

创建了数据库和数据表后, MySQL 也就具备了存储数据的能力。这样 MySQL 可在表中插入、修改和删除数据记录, 进而可使用查询语句查询数据表中的数据记录。

1. 插入数据记录

创建表后, 需要向表中插入数据记录。系统管理员可通过 load data 和 insert 语句完成插入数据记录的任务。

(1) load data 语句的格式为:

```
load data local infile 数据文件的绝对路径 into table 数据表名;
```

其中 load data 命令是表示将文件中的数据插入到数据表中; local infile 表示数据文件来源于本地磁盘; 数据文件的绝对路径指向按照一定格式编写的数据记录数据文件; into table 选项与紧跟其后的数据表名表示将数据文件中的数据插入到指定的数据表中。

数据文件的编写格式要求为: 每一行代表一条数据记录; 每一行的数据字段值的顺序严格按照指定的数据表中数据字段的顺序进行排列; 每一行的数据字段值使用一个 Tab 制表符分隔; 若有空值, 使用 “\N” 代替。

(2) insert 语句的语法格式为:

```
insert into 表名[(字段名1, 字段名2, …)] value(字段1的值, 字段2的值, …);
```

其中 insert 是 SQL 语言中插入数据的关键字; into 表示指向数据表; 表名指定所有要操作的数据表名称; 字段名 1…指定数据表中需要插入数据的字段; 字段 1 的值…指定与字段名相对应的数据。

例 19-25: 从数据文件中添加数据记录

将数据文件/home/dranyu/Documents/userInfo_data.txt 的数据添加到 web3ds 数据库的 userInfo 数据表中。userInfo_data.txt 数据文件的数据如图 19-3 所示。数据是按照 userInfo 数据表的字段顺序 (userId userName sex birthday) 排列, 每个字段值以一个 Tab 制表符隔开。

数据文件准备好后, 选择 web3ds 数据库, 执行命令和结果显示如下所示:

```
mysql> load data local infile "/home/dranyu/Documents/userInfo_data.txt" into table
userInfo;
Query OK, 3 rows affected (0.00 sec) //3行被影响, 表示数据文件中3条数据添加到了表中
Records: 3 Deleted: 0 Skipped: 0 Warnings: 0 //Records:3表示3条数据记录
```



图 19-3 userInfo 数据文件

数据插入完成后，可使用 select 语句查询 userInfo 所存储的数据记录，显示 userInfo 数据表中的所有数据记录，命令执行和结果显示如下所示：

```
mysql> select * from userInfo; //查询 userInfo 表的所有数据记录
+-----+-----+-----+
| userId | userName | sex | birthday | //userInfo 所有字段名
+-----+-----+-----+
| dranyu | dranyu.pan | b | 1984-10-28 | //第 1 条数据记录
| sunny | sunny.hang | g | 1985-03-05 | //第 2 条数据记录
| jum | jum.suse | g | NULL | //第 3 条数据记录，数据文件中的 "\N" 变为了 NULL
+-----+-----+-----+
3 rows in set (0.00 sec)
```

补充说明

数据记录查询语句的具体使用在下小节讲述。select * from userInfo 表示查询并显示 userInfo 数据表中的所有数据记录。

例 19-26：插入单条完整的数据记录

在使用 insert 语句插入数据记录时，若需要输入的数据记录的数据值顺序与表的字段顺序一致，可省略指定表的字段名选项。向 userInfo 插入一个完整的数据记录，命令执行和结果显示如下所示：

```
mysql> insert into userInfo values('tom','tom.tomas','b','1989-12-12'); //插入数据记录
Query OK, 1 row affected (0.00 sec)
//显示数据记录。
mysql> select * from userInfo;
+-----+-----+-----+
| userId | userName | sex | birthday | //新添加的数据记录
+-----+-----+-----+
| tom | tom.tomas | b | 1989-12-12 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

例 19-27：插入指定字段的数据记录

有时一条数据记录中会出现多个字段的值都为空，这时若使用例 19-26 的方式添加数据记录，工作效率很低。这时可使用指定插入数据记录字段的方式添加数据记录。向 userInfo 数据表添加数据记录 sasa sasa.chen g null (null 表示 birthday 字段的值为空)，命令执行和结果显示如下：



```
mysql> insert into userInfo(userId,userName,sex) values('sasa','sasa.chen','g'); //  
没有指定  
birthday 字段  
Query OK, 1 row affected (0.00 sec)  
//显示 userInfo 数据表的所有数据记录  
mysql> select * from userInfo;  
+-----+-----+-----+-----+  
| userId | userName | sex | birthday |  
+-----+-----+-----+-----+  
| sasa | sasa.chen | g | NULL | //系统自动为 birthday 字段添加了 NULL 空值  
+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

补充说明

对于没有输入值的字段，MySQL 数据库管理系统会自动为其添加一个值。若此字段设置了默认值，系统会将此字段的值设置为默认值；否则设置为 NULL 空值。

2. 查询数据记录

数据保存在数据表中，在使用数据的时间需要使用 select 语句从数据表中读取数据。语句的基本格式为：

```
SELECT 表的字段列表 FROM 表名 WHERE 条件表达式;
```

其中表的字段列表是指定数据表中字段名的排列或匹配符“*”，每个字段名使用“,”隔开，如 userId,userName；查询条件表达式可是逻辑表达式，如 userId='dranyu'，也可是字符串表达式，如 birthday IS NOT NULL 等。

例 19-28：查询所有数据记录

查询 userInfo 数据表的所有记录，命令执行和结果显示如下：

```
mysql> select * from userInfo; //输入命令，按下回车键  
+-----+-----+-----+-----+  
| userId | userName | sex | birthday |  
+-----+-----+-----+-----+  
| dranyu | dranyu.pan | b | 1984-10-28 |  
| sunny | sunny.hang | g | 1985-03-05 |  
| jum | jum.suse | g | NULL |  
| tom | tom.tomas | b | 1989-12-12 |  
| sasa | sasa.chen | g | NULL |  
+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

例 19-29：显示指定的数据列

查询并显示 userInfo 数据表中 userName 和 sex 字段的数据，命令执行和结果显示如下所示：

```
mysql> select userName,sex from userInfo;  
+-----+-----+  
| userName | sex | //显示了 userInfo 中 userName 列和 sex 列的数据  
+-----+-----+  
| dranyu.pan | b |  
| sunny.hang | g |  
| jum.suse | g |  
| tom.tomas | b |  
| sasa.chen | g |  
+-----+-----+
```

```
5 rows in set (0.00 sec)
```

例 19-30：使用逻辑表达式显示数据记录

显示 userInfo 数据表中 userId 为 dranyu 的数据记录，命令执行和结果显示如下所示：

```
mysql> select * from userInfo where userId='dranyu';
+-----+-----+-----+
| userId | userName | sex | birthday |
+-----+-----+-----+
| dranyu | dranyu.pan | b   | 1984-10-28 | //数据记录
+-----+-----+-----+
1 row in set (0.00 sec)
```

例 19-31：使用字符串表达式显示数据记录

显示 userInfo 数据表中 birthday 不为空的数据记录，命令执行和结果显示如下所示：

```
mysql> select * from userInfo where birthday is not null;
+-----+-----+-----+
| userId | userName | sex | birthday |
+-----+-----+-----+
| dranyu | dranyu.pan | b   | 1984-10-28 |
| sunny  | sunny.hang | g   | 1985-03-05 |
| tom    | tom.tomas | b   | 1989-12-12 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

补充说明

数据的查询是 MySQL 数据库使用过程中最基础也是最重要的功能。数据查询语句很简单，也会很复杂。简单时，查询语句只有一句话；复杂时，可长达几十行甚至几百行。更深入学习可上 MySQL 的官方网站 (<http://dev.mysql.com/doc/refman/5.1/zh/tutorial.html>)。

3. 删除数据记录

删除数据表中的数据记录可使用 delete 语句，格式为：

```
delete from 表名 where 条件表达式;
```

例 19-32：删除 userInfo 表中，userId 为 tom 的记录，命令执行和结果显示如下所示：

```
mysql> delete from userInfo where userId='tom'; //userId='tom'是条件语句
Query OK, 1 row affected (0.00 sec)
//显示 userInfo 表所有数据记录
mysql> select * from userInfo;
+-----+-----+-----+
| userId | userName | sex | birthday |
+-----+-----+-----+
| dranyu | dranyu.pan | b   | 1984-10-28 |
| sunny  | sunny.hang | g   | 1985-03-05 |
| jum   | jum.suse   | g   | NULL      |
| sasa  | sasa.chen | g   | NULL      |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

只要数据记录满足条件表达式，数据记录就会被删除。

例 19-33：删除 userInfo 表中 birthday 字段值为 NULL 空值的所有数据记录，命令执行和结果显示如下所示：

```
mysql> delete from userInfo where birthday is null; //birthday is null是条件表达式
Query OK, 2 rows affected (0.00 sec)
```



```
//显示删除数据记录后 userInfo 表的所有数据记录
mysql> select * from userInfo;
+-----+-----+-----+
| userId | userName | sex | birthday |
+-----+-----+-----+
| dranyu | dranyu.pan | b   | 1984-10-28 |
| sunny  | sunny.hang | g   | 1985-03-05 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

在不使用任何的表达式时, delete 语句会将指定数据表的所有数据记录删除。

例 19-34: 删除 userInfo 表中所有的数据记录, 命令执行和结果显示如下所示:

```
mysql> delete from userInfo; //删除所有数据记录
Query OK, 2 rows affected (0.00 sec)
//显示执行清除命令后 userInfo 表的所有数据记录
mysql> select * from userInfo;
Empty set (0.00 sec)
```

4. 修改数据记录

修改数据记录也是 MySQL 数据库管理重要的一部分。可使用 update 语句修改数据表中的数据记录, 其基本格式为:

```
update 表名 set 字段名 1=字段值 1, [字段名 2=字段值 2, …] where 条件表达式;
```

例 19-35: 修改 userInfo 表中 userId 为 sasa 的数据记录里的 birthday 字段值, 命令执行和结果显示如下所示:

```
mysql> update userInfo set birthday='1982-4-2' where userId='sasa';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0 //表示查找到 1 条记录, 已修改 1 条记录,
          数据修改成功
mysql> select * from userInfo; //显示 userInfo 表的所有数据记录
+-----+-----+-----+
| userId | userName | sex | birthday |
+-----+-----+-----+
| sasa   | sasa.chen | g   | 1982-04-02 | //数据已修改
+-----+-----+-----+
5 rows in set (0.00 sec)
```

补充说明

同样, update 语句可修改一条数据记录, 也可修改多条数据记录。在使用 update 语句时需要注意一定要添加 where 条件限制。如果没有 where 条件限制, update 语句将修改整张数据表的相应字段值。这就有可能导致大量数据被破坏。

19.3.5 数据库用户管理

MySQL 应用程序在安装的时候就默认添加了 root 用户, 一个数据库管理员不能很好地管理数据库, 有时需要多几个 MySQL 的用户, 这样可方便数据库的管理而且更加安全。在 MySQL 数据库管理系统中管理用户需要使用到 GRANT 和 REVOKE 命令。

1. GRANT 命令添加用户并授予权限

使用 GRANT 命令可添加用户和授予相关的访问权限, GRANT 命令的基本语法为:

```
GRANT 权限类型 ON 对象 TO 用户 [IDENTIFIED BY 用户密码]
```

其中权限类型包括 USAGE（无权限）、SELECT、CREATE、UPDATE 和 DELETE 等；对象表示操作的对象包括数据表、函数和存储过程等。“IDENTIFIED BY 用户密码”设置用户的密码。

例 19-36：添加一个名为 dranyu，密码为 123456 的用户，允许 dranyu 用户从本地登录，具体步骤如下：

(1) 只有 root 用户登录了 MySQL 数据库管理系统才能添加用户，root 用户登录的命令如下：

```
dranyu@dranyu-desktop:~$ mysql -u root -p
Enter password: //输入 root 的 MySQL 数据库管理系统的密码
mysql>
```

(2) 添加一个没有任何权限，名为 dranyu，密码为 123456 的用户，命令如下所示：

```
mysql> GRANT USAGE ON *.* //键入回车键
      -> TO dranyu@localhost IDENTIFIED BY "123456";
Query OK, 0 rows affected (0.00 sec)
```

USAGE 表示设置用户无权限访问；*.*表示 MySQL 数据库管理系统所有对象，第 1 个*表示所有的数据库，第 2 个*表示相应数据库中的所有对象；dranyu@localhost 是用户名 dranyu 在 localhost 主机登录的名称；IDENTIFIED BY “123456”表示将 dranyu 用户的登录密码设置为 123456。此时，用户 dranyu 可登录 MySQL 数据库管理系统。使用 dranyu 登录 MySQL 数据库，选择 MySQL 数据库，命令执行和结果显示如下所示：

```
dranyu@dranyu-desktop:~$ mysql -u dranyu -p //以 dranyu 身份登录
Enter password: //输入 dranyu 的用户密码 123456
mysql> use web3ds; //选择 MySQL 数据库
ERROR 1044 (42000): Access denied for user 'dranyu'@'localhost' to database 'web3ds'
```

在选择 MySQL 数据库时，显示了 ERROR 1044 表示 dranyu@localhost 登录用户没有权限访问 MySQL 数据库。

例 19-37：修改用户的权限

使用户 dranyu@localhost 拥有对 web3ds 数据库所有对象的查询 SELECT、插入 INSERT、创建 CREATE、删除 DELETE 和修改 UPDATE 权限，命令执行和结果显示如下所示：

```
mysql> GRANT SELECT,INSERT,CREATE,DELETE,UPDATE ON web3ds.* 
      -> TO dranyu@localhost;
Query OK, 0 rows affected (0.00 sec)
```

使用 dranyu 用户再次登录到 MySQL 服务器上，选择 web3ds 数据，就可管理 web3ds 数据库中的表和数据。

补充说明

MySQL 数据库管理系统中的用户信息都保存在 MySQL 数据库中的 user 表中，系统管理员可进入 MySQL 数据库后使用 select * from user;语句查询 user 表中的数据记录。

2. REVOKE 命令撤销权限

使用 REVOKE 命令撤销 MySQL 用户的相关权限。REVOKE 命令的基本格式如下：

REVOKE 权限类型 ON 对象 FROM 用户

其中所使用的参数含义与 GRANT 命令中参数含义相同。



例 19-38：撤销 dranyu 用户对 web3ds 数据库的创建 CREATE 和修改 UPDATE 数据的权限，命令如下。

```
mysql> REVOKE CREATE,UPDATE ON web3ds.* //撤销 create 和 update 的权限  
      -> FROM dranyu@localhost;  
Query OK, 0 rows affected (0.00 sec)
```

例 19-39：撤销 dranyu 用户对 web3ds 数据库的所有权限，命令执行和结果显示如下所示：

```
mysql> REVOKE ALL PRIVILEGES ON web3ds.* //撤销 dranyu 用户对 web3ds 数据库  
的所有权限  
      -> FROM dranyu@localhost;           //指定撤销权限的用户 dranyu@localhost  
Query OK, 0 rows affected (0.01 sec)
```

19.3.6 数据库的备份和恢复

在数据库管理中，对数据进行定时备份是十分重要的。MySQL 应用程序提供了一个 mysqldump 命令，帮助系统管理员备份数据库。mysqldump 命令的基本语法格式为：

```
mysqldump [选项] [数据库] [表]  
mysqldump 常用的选项如下。
```

- -u 用户名：指定操作备份的用户名。
- -p：指定用户的密码。
- -d：表示导出数据库的结构而不导出数据。
- -t：表示导出数据而不导出结构。

例 19-40：以 root 的身份将 web3ds 数据库的结构和数据导出到 /home/dranyu/web3ds_data.sql 文件，命令执行如下所示：

```
root@dranyu-desktop:~# mysqldump -u root -p web3ds >/home/dranyu/web3ds_data.sql  
Enter password: //输入 MySQL 服务器 root 的密码
```

例 19-41：以 root 的身份将 web3ds 数据库中 userInfo 表的表结构导出到 /home/dranyu(userInfo.sql，命令执行如下所示：

```
root@dranyu-desktop:~# mysqldump -u root -p -d web3ds userInfo  
>/home/dranyu/userInfo.sql  
Enter password: //输入 MySQL 服务器 root 的密码
```

由于 mysqldump 导出数据库的文件格式是 sql 文件格式，这样 MySQL 应用程序可很好地解释这些导出的文件。系统管理员可使用 MySQL 命令导入由 mysqldump 导出的数据库或表。

例 19-42：导入整个数据库

使用例 19-39 导出的备份文件 web3ds_data.sql，将备份数据库导入到一个新建的数据库 web3ds2 中，具体步骤如下所示：

(1) 创建一个新的数据库 web3ds2，命令执行和结果显示如下所示：

```
root@dranyu-desktop:~# mysqladmin -u root -p create web3ds2 //以 root 身份创建一个  
web3ds2 数据库  
Enter password: //输入 MySQL 服务器 root 的密码
```

(2) 将备份文件 web3ds_data.sql 里的表结构和数据导入到新建的 web3ds2 数据库中，命令执行和结果显示如下所示：

```
//以 root 的身份将 /home/dranyu/web3ds_data.sql 导入 web3ds2 数据库  
root@dranyu-desktop:~# mysql -u root -p web3ds2 </home/dranyu/web3ds_data.sql  
Enter password: //输入 MySQL 服务器 root 的密码
```

(3) 以 root 身份进入 MySQL 服务器的字符操作界面, 检查使用了 web3ds2 数据库的创建, 结果显示如下所示:

```
root@dranyu-desktop:~# mysql -u root -p
Enter password: //输入 MySQL 服务器 root 的密码
mysql> show databases; //显示服务器上所有的数据库
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| web3ds         |
| web3ds2        | //新添加的 web3ds2 数据库
+-----+
4 rows in set (0.00 sec)
mysql> use web3ds2    //选择 web3ds2 数据库
Database changed
mysql> show tables;   //显示 web3ds2 数据库的所有表
+-----+
| Tables_in_web3ds2 |
+-----+
| userInfo          |
| vip               |
| vipInfo           |
+-----+
3 rows in set (0.00 sec)
mysql> select * from userInfo; //查询数据表 userInfo 中的数据
+-----+-----+-----+
| userId | userName | sex   | birthday |
+-----+-----+-----+
| dranyu | dranyu.pan | b    | 1984-10-28 |
| sunny  | sunny.hang  | g    | 1985-03-05 |
| jum   | jum.suse   | g    | NULL     |
| tom   | tom.tomas  | b    | 1989-12-12 |
| sasa  | sasa.chen  | g    | 1982-04-02 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

通过测试后, 可确定 web3ds2 数据库是 web3ds 数据库的一个复制数据库, 与 web3ds 数据库有相同的数据表结构和数据。

19.4 小结



MySQL 数据库是一个关系型数据库, 其功能强大, 占用系统资源少, 数据查询效率高, 成为中小型企业的首选, 现在也有比较多的大企业转为使用 MySQL 数据库作为企业的数据管理的软件。

本章讲述了在 Ubuntu 如何安装、配置和使用 MySQL 数据库。通过数据库的操作实例简述了使用 create、drop 等命令管理数据库, 使用 insert、delete 和 update 等语句管理数据表中的数据记录。为了系统管理员可方便地 MySQL 服务器, 合理 MySQL 服务器, 本章介绍了



MySQL 服务器的用户管理。为了数据库中数据的安全，本章解释了如何对数据库进行备份和恢复。

19.5 习题



1. 简述 MySQL 的特征。
2. 完成安装 MySQL 服务器软件包的操作。
3. 将 MySQL 服务器主配置文件的内容修改为示范文件 my-medium.cnf 的内容。
4. 完成关闭和启动 MySQL 服务器的操作。
5. 使用 mysqladmin 命令修改 root 用户的密码。
6. 使用 mysql 命令进入 MySQL 字符操作界面，选择 MySQL 数据库，使用 select 查询语句查询 user 表的内容。
7. 在 MySQL 字符操作界面上创建一个名为 web3d 的数据库，然后显示服务器上所有的数据库信息验证是否添加成功。
8. 使用 create table 语句创建如下所示表结构名为 user 的数据表。

```
表 user (userId,username,sex,birthday,phone,homeAddr,mark)
userId char(18) not null,
userName varchar(20) not null,
sex char(1) default 'b',
birthday date,
phone varchar(20),
primary key (userId)
```

9. 将以下内容以数据文件的格式添加文件 data.txt 中，然后将数据文件 data.txt 的数据插入到习题 8 所创建的 user 数据表中。

```
dranyu    dranyu.pan    b    1984-10-28  13254694562
sunny     sunny.hang    g    1985-03-05  12356569854
jum      jum.suse      g \N   12456667889
```

10. 使用 GRANT 命令创建一个名为 sunny 的用户，允许它在本地主机上登录，设置其对 web3d 数据库中的所有表拥有查看数据记录、插入数据记录和修改数据记录的权限。
11. 在习题 10 的基础上，使用 REVOKE 命令撤销 sunny 用户对 web3ds 数据库中所有表插入数据记录和修改数据记录的权限。
12. 使用 mysqldump 命令将 web3d 数据库导出到 web3d_data.txt 文件，然后创建一个新的数据库 web3ds2，将 web3ds_data.txt 文件导入到 web3ds2 数据库中。

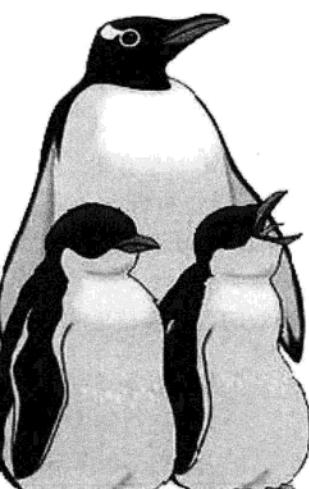
Linux

第7部分 Linux 编程

第20章

编程工具

以前，Linux 编程都是在命令行下进行的，只有编程高手才能在 Linux 系统下进行编程。目前，随着图形操作界面在 Linux 系统上的使用，越来越多的开发者在 Linux 系统环境下进行软件开发的工作。本章讲述 Ubuntu 环境下 C/C++的编程环境，主要介绍如何使用 Ubuntu 下的集成开发环境，以及如何使用常用命令工具。





20.1 Ubuntu 下编程环境概述

Ubuntu 的目标就是为用户提供一个简单易用的免费开源的操作系统。无论是办公还是娱乐，抑或是开发软件，用户同样可使用 Ubuntu 提供的图形操作界面轻松完成。Ubuntu 保留了 Linux 下命令操作界面的编程方式。同时，也结合了目前广泛使用的可视化编程工具。

20.1.1 编程工具简介

Linux 系统中提供了许许多多的编程工具，其中包括如下。

- 编辑工具：Linux 下常用的编辑工具包括了 Gedit 文本编辑器、vim 编辑器和 emacs 代码编辑器。这些编辑工具的使用，使程序的编辑与编译工作分开。
- 编译工具：编译工具将程序员编写的代码翻译为计算机可识别并可执行的代码。在 Linux 下，常用的 C/C++ 编译工具包括 GCC 和 G++。
- 配置文件生成工具：一个大型软件总是由多个源程序组成的，为了能够将大量的编译命令做一次完成，Linux 中提供了 make 和 autoconf 工具，分别用于大型软件的编译和编译前根据机器当前状态作相应配置。
- 可视化编程开发工具：可视化编程开发工具简化了程序员开发流程，使程序员将更多的精力放在编写代码上。Ubuntu Linux 下常用的 C/C++ 可视化编程开发工具包括 Anjuta 和 Eclipse CDT。
- 开发工具包：Linux 提供了优秀的 GNU C 函数库、Motif 函数库、GTK 函数库和 QT 函数库等。
- 项目管理工具：Ubuntu 还提供了 CVS 和 SVN 优秀的版本控制软件。

补充说明

有关 CVS 和 SVN 的知识在后续章节中讲述。

20.1.2 Ubuntu 下的编程类型

Ubuntu 下包括了多种编程类型，常见类型如下所示：

(1) shell 编程：shell 是 Linux 操作系统的最外层，直接与用户交互，负责解释命令。就像 DOS 中的 COMMAND.COM 程序。shell 编程与编写.BAT 文件类似，但其提供了一整套机制，使其具有很强大的功能。善于使用 Shell 编程可以更好地使用和管理 Linux 操作系统。

(2) WEB 编程：Ubuntu Linux 系统拥有最好、最普及的 WEB 服务器 Apache，还有优秀的 PHP 脚本语言。WEB 编程包括 PHP、JSP、Perl 的开发编程。

(3) 普通程序开发：在 Ubuntu Linux 系统中，使用 C/C++ 语言开发出高效的、可移植性的应用程序。

(4) X-Win 编程：在 Ubuntu Linux 系统下，可使用 MOTIF 函数库、QT 函数库和 KDE 函数库等开发从 UNIX 系统继承过来的图形界面系统 X-Window，以及其环境下的应用程序。

20.2 常用编辑器



Ubuntu 默认提供了 Gedit 文本编辑器，但在编程中很少有程序员会选择它作为代码编辑器。程序员普遍使用 vim 或 emacs 等专业的程序编辑器。vim 可与终端完美地结合，能快速切换代码的编辑和编译工作。emacs 是一个具有很强扩展性的免费文本编辑器，可编写代码、收发邮件、登录新闻组、访问 Internet。

20.2.1 VIM 编辑器

vi 是 Linux 系统的第一个全屏幕交互式编辑程序，它从诞生至今一直得到广大用户的青睐，历经数十年仍然是人们主要使用的文本编辑工具，足见其生命力之强，而强大的生命力是其强大的功能带来的。

1. vi/vim 简介

Vi 是“Visual interface”的简称，它在 Linux 上的地位就仿佛 Edit 程序在 DOS 上一样。它可以执行输出、删除、查找、替换、块操作等众多文本操作，而且用户可以根据自己的需要对其进行定制，这是其他编辑程序所没有的。vi 不是一个排版程序，它不像 Word 或 WPS 那样可以对字体、格式、段落等其他属性进行编排，它只是一个文本编辑程序。vi 没有菜单，只有命令，且命令繁多。vi 有 3 种基本工作模式：命令模式、输入模式和末行命令模式。

各种模式之间可以相互切换，vi 启动的时候是进入到命令模式的。从命令模式进入到文本输入模式，可以通过输入“i”，“I”，“o”，“O”，“a”，“A”，“s”和“S”。而从文本输入模式中进入到命令模式，只要按下“Esc”退出就可以了。若从命令模式切换到末行命令模式，输入“shift+：“符号即可。在末行模式中命令执行完毕之后，就会回到命令模式中。

在 Ubuntu 9.04 中，系统默认安装 vim-tiny。vim-tiny 软件包是一个简化版本，其加载速度快，基本上满足用户的需求，具备日常文本处理的功能。若用户喜欢使用 vim 版本，可以在新立得软件包管理器中添加 vim，或是使用 apt-get install vim 安装 vim。本书中主要使用 vim 来进行讲解。

2. 基本操作

vim 的基本操作包括了文件的创建、文件内容的输入、文件保存和关闭文件。为了保持操作的连续性，通过一个完整的操作步骤说明 vim 的基本操作。

例 20-1：vim 的基本操作

创建一个新文件，输入文件内容，将文件保存为 hello.c，退出 vim 编辑器，步骤如下所示：

(1) 在终端输入 vim 命令然后按下回车键执行命令，进入 vim 的编辑界面如图 20-1 所示。



图 20-1 vim 开始界面

(2) 进入 vim 后, vim 处于命令模式。当前文件是空文件, 而且没有名称。为了给文件保存并命名, 按下“Shift+：“组合键, 进入了末行命令模式, 在提示符后输入“w hello.c”命令, 如图 20-2 所示。然后按下回车键, 执行保存命令。命令执行完毕之后, 同时回到 vim 命令模式。

(3) 此时文件 hello.c 已保存。为了编辑此文件按下插入文本的命令, 如 i 键, 进入到输入模式。此时, 在底部显示如图 20-3 的结果, 表示进入了输入模式。



图 20-2 输入末行命令

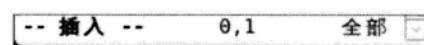


图 20-3 进入输入模式后的提示

(4) 进入输入模式后, 输入文件的内容如图 20-4 所示。



图 20-4 输入文件内容

(5) 输入内容完毕之后, 按下 Esc 键, 退出输入模式, 同时进入了 vim 的命令模式。按下“Shift+：“组合键进入末行命令模式, 在末行提示符“：“后输入 wq 命令, 按下回车键保

存文件并关闭文件，退出 vim 程序。

补充说明

此基本操作说明了 vim 中三种模式之间的切换方法，同时使用一些命令模式和末行命令模式下的命令。**i** 键是命令模式下的命令，其表示进入输入模式并且从光标当前位置开始输入文件。**w** 键是末行命令模式下的命令，其表示保存文件。**q** 键是末行命令模式下的命令，其表示关闭文件并退出 vim。各命令的详细信息在以下小节讲述。

3. 命令模式下的命令

vim 工具在命令模式下的命令可分为移动光标类、屏幕翻滚类、插入文本类、文件修改类和搜索文本类等几大类。移动光标类的命令主要功能是在命令模式下快速移动光标，将光标移动到指定位置，常用的命令如表 20-1 所示。

表 20-1 移动光标类命令

命 令	说 明
H	将光标左移一个字符，在同一行中移动
I	将光标右移一个字符，在同一行中移动
space	将光标右移一个字符，在同一段中移动
Backspace	将光标左移一个字符，在同一段中移动
k 或 Ctrl+P	将光标上移一行
j 或 Ctrl+N	将光标下移一行
Enter	将光标下移一行
w 或 W	将光标右移至下一个字的字首，使用 w 标点符号算一个字，使用 W 时移动是以空格和分行符为移动的标记
b 或 B	将光标左移至下一个字的字首，使用 b 标点符号算一个字，使用 B 时移动是以空格和分行符为移动的标记
e 或 E	将光标右移至下一个字的字首，使用 e 标点符号算一个字，使用 E 时移动是以空格和分行符为移动的标记
)	将光标移至句尾
(将光标移至句首
}	将光标移至段落开头
{	将光标移至段落结尾
nG	将光标移至第 n 行首。如 2G 表示将光标移动到第 2 行行首
n+	将光标下移 n 行
n-	将光标上移 n 行
n\$	将光标移至第 n 行尾
H	将光标移至屏幕顶行
M	将光标移至屏幕中间行
L	将光标移至屏幕最后行
0	将光标移至当前行首
\$	将光标移至当前行尾



在 vim 的编程字符界面上，信息是一屏一屏显示的，这就要求 vim 应具有屏幕翻滚的功能。vim 提供屏幕翻滚类命令使开发人员可方便滚屏，常用命令如表 20-2 所示。

表 20-2 屏幕翻滚类命令

命 令	说 明
Ctrl+U	向文件首翻半屏
Ctrl+D	向文件尾翻半屏
Ctrl+F	向文件尾翻一屏
Ctrl+B	向文件首翻一屏
nz	将第 n 行滚至屏幕顶部，不指定 n 时将当前行滚至屏幕顶部

只有在命令模式执行插入内容类命令，vim 才能从命令模式切换到输入模式。插入内容类的命令如表 20-3 所示。

表 20-3 插入内容类命令

命 令	说 明
i	在当前光标前开始输入
I	在当前光标所在行的行首开始输入
a	在当前光标后开始输入
A	在当前光标所在行的行尾开始输入
o	在当前行之下新开一行开始输入
O	在当前行之上新开一行开始输入
r	替换当前光标所在位置的字符
R	替换当前字符及其后的字符，直至按 ESC 键
s	从当前光标位置处开始，以输入的文本替代指定数目的字符
S	删除指定数目的行，并以所输入文本代替之
nw 或 nCW	修改指定数目 n 的字
nCC	修改指定数目 n 的行

补充说明

使用 R 命令后，vim 进入替换的输入模式；执行 r 命令后，用户只能输入一个字符修改光标当前所在的字符，修改完毕之后，vim 回到命令模式。

在输入模式下，程序员可直接使用 Backspace 退格键删除字符。同时，vim 也提供在命令模式下运行的删除指定文本、复制文本、粘贴文本和撤销等命令，如表 20-4 所示。

表 20-4 文本修改类命令

命 令	说 明
X 或 x	删除一个字符，x 表示删除当前光标所在位置的字符，X 表示删除光标前的一个字符
nX 或 nx	删除 n 个字符，nx 表示删除当前光标所在位置后 n 个的字符，X 表示删除光标前 n 个字符

续表

命 令	说 明
dd	删除光标所在的行
ndd	删除从光标所在的行开始往下的 n 行
d0 或 d\$	d0 表示删除当前光标位置前一个字符到行首的所有字符, d\$ 表示删除当前光标位置到行末的所有字符
cw	更改光标所在处的字符到字尾处的所有字符
cnw	更改光标所在字符后 n 个字, 第一个字的获取方式与 cw 命令的相同
yw	将光标所在之处到字尾的字符复制到缓冲区中
nyw	复制 n 个字到缓冲区, 第一个字的获取方式与 yw 命令的获取文件同, 之后的每个字以空格或换行符分开, 而且标点符号为一个字
yy	复制光标所在行到缓冲区
nyy	复制光标所在行到第 n 行的文本内容到缓冲区。如 4yy 表示将当前行开始到第 4 行的文本内容复制到缓冲区
p	将缓冲区内的字符贴到光标所在位置。注意: 所有与“y”有关的复制命令都必须与“p”配合才能完成复制与粘贴功能
u	撤销上一命令的修改操作。按多次 u 命令可以执行多次撤销

vim 提供了对文本内容的搜索和替换功能, 常用命令如表 20-5 所示。

表 20-5 搜索文本类命令

命 令	说 明
/pattern	从光标开始处向文件尾搜索第一个 pattern
?pattern	从光标开始处向文件首搜索第一个 pattern
n	在同一方向重复上一次搜索命令
N	在反方向上重复上一次搜索命令

4. 末行命令模式下的命令

vim 提供了强大的末行命令功能, 在末行命令模式下, 程序员可使用正则表达式完成文件内容的批量复制、删除和移动。末行命令模式下常用的命令如表 20-6 所示。

表 20-6 搜索文本类命令

命 令	说 明
:n1,n2 co n3	将 n1 行到 n2 行之间的内容复制到第 n3 行下
:n1,n2 m n3	将 n1 行到 n2 行之间的内容移至第 n3 行下
:n1,n2 d	将 n1 行到 n2 行之间的内容删除
:s/p1/p2/g	将当前行中所有 p1 均用 p2 替换
:n1,n2s/p1/p2/g	将第 n1 至 n2 行中所有 p1 均用 p2 替换
:g/p1/s/p2/g	将文件中所有 p1 均用 p2 替换
:w	保存当前文件
:e filename	打开文件 filename, 并编辑此文件



续表

命 令	说 明
:x	保存当前文件并退出
:q	退出 vim 应用程序
:q!	不保存文件并强制退出 vim 应用程序
:!command	执行 shell 的 command 命令
:n1,n2 w!command	将文件中 n1 行至 n2 行的内容作为 shell 下 command 命令的输入并执行之, 若不指定 n1, n2, 则表示将整个文件内容作为 command 命令的输入

20.2.2 Emacs 编辑器

Emacs 编辑器由 Lisp 编程语言编写, 具有很强的扩展性。正是它强大的扩展性, 使它的功能早已超越了一个文本编辑器。

1. Emacs 简介

Emacs 是目前世界上最具可移植性的重要软件之一, 能够在当前大多数操作系统上运行, 包括类 UNIX 系统 (GNU/Linux、各种 BSD、Solaris、AIX、IRIX、Mac OS X 等)、MS-DOS、Microsoft Windows, 以及 OpenVMS 等。

Emacs 拥有跨平台性、文本编辑功能强大、高度可定制性、高度可扩展性和同时支持字符界面与图形界面两种方式等优点。其主要的功能和组件如下所示:

- 强大的文本编辑功能

Emacs 支持多种编程语言如 c/c++/java/sql/shell 等; 支持语法高亮、代码缩进、多种形式的代码折叠和块操作等, 还可以通过安装插件支持新出现的编程语言。

- GCC、GDB、CC、DBX

Emacs 内置以上四种编译器、调试器的接口, 为用户提供了一个可视化的编译、调试环境。无论是编译还是调试时都可以方便定位到想看的源码上。

- ECB

ECB 是一个为 Emacs 编辑器而开发的文件和代码浏览器, 支持 Java、C/C++、ELisp、Perl 等源码文件。

- WOMAN

Emacs 集成的查看 man page 的工具, 比终端的 man 命令工具要方便, 此工具名为 WOMAN, 其全称是 With Out Man。

2. Emacs 基本操作

在默认情况下, Ubuntu 没有安装 Emacs 应用程序。安装 Emacs 应用程序使用如下命令:

```
root@dranyu-desktop:~# apt-get install emacs22
```

安装 Emacs22 软件包后, 就可使用 Emacs 应用程序编辑代码。在终端输入 Emacs 命令启动 Emacs 应用程序, 启动后, 它的主界面如图 20-5 所示。

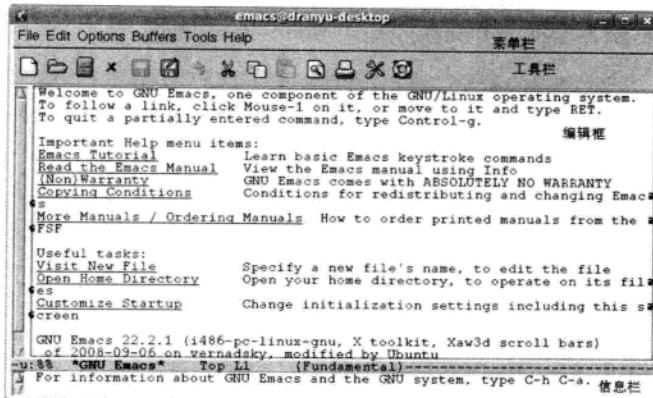


图 20-5 Emacs 主界面

在主界面上，使用鼠标选择编辑框中的“Emacs Tutorial”选项可查看 Emacs 的基本用法。Emacs 功能强大，为说明它的使用，以详细的基本操作步骤讲述其应用，如例 20-2 所示。

例 20-2：Emacs 的基本操作

使用 Emacs 应用程序创建一个 main.c，然后完成输入文本内容、保存文件、编译代码文件，具体步骤如下所示：

(1) 在终端输入 Emacs 命令，打开 Emacs 应用程序，单击工具栏上的新建文件按钮□，在制作界面的信息栏上显示终端的当前路径，如图 20-6 所示，并要求输入新建文件的名称。



图 20-6 输入新建文件的名称

(2) 输入 main.c 文件名，然后按回车键保存此文件。在此文件中，输入 c 程序代码内容，Emacs 高亮显示代码，编辑结果如图 20-7 所示。



图 20-7 输入代码

(3) Emacs 集成了 GCC 的编译器。选择“Tools”|“Compile”菜单命令，信息栏上显示



输入编译命令，在此输入 gcc -o main main.c，如图 20-8 所示。

```
-u:*** main.c All L5 (C/l Abbrev) -----  
I Compile command: gcc -o main main.c
```

图 20-8 输入编译命令

(4) 按下回车键执行命令。gcc 命令表示编译 main.c 文件，并生成可执行文件 main。命令执行完后，在 main.c 的同一目录下会被添加了一个 main 可执行文件。信息栏显示 gcc 命令的执行结果如图 20-9 所示。

```
-*- mode: compilation; default-directory: "~/D  
* Documents/program/" -*-  
Compilation started at Sat Dec 19 20:44:39  
gcc -o main main.c  
Compilation finished at Sat Dec 19 20:44:39  
-u:*** *compilation* All L1 (Compilation: exi
```

图 20-9 编译 main.c 文件结果

(5) 在 Emacs 应用程序中可直接运行 shell 命令。选择“Tools”|“Shell Command”菜单命令，在末端上，显示“shell command:”字样，在 shell command 后输入./main，如图 20-10 所示，表示运行 main 可执行文件。

```
I Shell command: ./main
```

图 20-10 输入 shell 的命令

(6) 按下回车键，执行命令。main 执行后输入结果如图 20-11 所示。

```
I Hello world!
```

图 20-11 执行 main 文件的结果

(7) 关闭文件。选择“File”|“Close”菜单命令关闭 main.c 文件，或者单击工具栏上的关闭按钮 关闭文件。

(8) 退出 Emacs 应用程序。选择“File”|“Exit Emacs”菜单命令退出 Emacs。

20.3 集成开发环境



集成开发环境，简称 IDE（Integrated Develop Environment），是用于提供程序开发环境的应用程序，一般包括代码编辑器、编译器、调试器和图形用户界面工具，就是集成了代码编写功能、分析功能、编译功能、调试功能等一体化的开发软件服务套装。Ubuntu 为 C/C++ 程序员提供了 Anjuta 和 Eclipse 两种常用而且得到广大程序员好评的集成开发环境。

PDG

20.3.1 Anjuta

Anjuta 是一个为 C/C++ 而开发的集成开发环境，为 Linux 和 UNIX 系列的各种命令行程序工具提供图形接口。其致力于解决复杂问题，并为功能强大的字符工具提供简单易用的 GNOME 图形用户接口。

1. Anjuta 的特点

Anjuta 应用程序是一个完全可定制的集成编辑器；实现了标签页与窗口共存；内嵌交互性强的源代码级别调试器 gdb；使用程序向导指引程序员快速创建各种项目；支持动态标记浏览方式，使程序员可快速定位到函数、结构和类的定义；提供了完整项目和编译文件的管理。

2. 安装 Anjuta

Anjuta 是开源的应用程序并且得到广大程序员的拥护，其加入到 Ubuntu 的软件库也就成了必然。安装 Anjuta 的方法包括在新立得软件包管理器中搜索 Anjuta 软件包并且安装此软件包和在终端上输入安装命令安装此软件包。在终端安装软件包所输入命令如下所示：

```
root@dranyu-desktop:~# apt-get install anjuta
```

安装 Anjuta 之后，在 Ubuntu 主菜单上就会添加了 Anjuta 应用程序的启动菜单“应用程序”|“编程”|“Anjuta IDE”命令，选择此菜单即可启动 Anjuta 应用程序。

3. Anjuta 基本操作

Anjuta 能创建多种编程语言的项目，如 C/C++、Java 等，而且功能强大。在本小节中创建一个 Hello world 的 c 语言应用程序并简述 Anjuta 的基本操作，如例 20-3 所示。

例 20-3：使用 Anjuta 创建、编写和编译 hello 项目

完成 Anjuta 创建编写和编译 hello 项目的具体步骤如下：

(1) 启动 Anjuta。选择“应用程序”|“编程”|“Anjuta IDE”菜单命令，启动 Anjuta 应用程序，其主界面如图 20-12 所示。



图 20-12 Anjuta 的主界面

(2) 创建工程。选择“文件”|“新建”|“工程”菜单命令，弹出 Anjuta 对话框，进入工程的创建引导程序，如图 20-13 所示。其中对话框上的选项卡上显示了 Anjuta 支持的编程语言，选择不同的选项卡，可相应地创建工程项目，在此选择 C 选项卡。



Ubuntu Linux 从初学到精通

(3) 在工程类型列表框中选择“Makefile project”选项，然后单击“前进”按钮，进入下一个界面“Project options”。

(4) 在设置工程属性界面上的“Project name”输入栏中输入新建工程的名称，如 hello。在“Project directory”输入栏中输入新建工程文件的保存位置，如~/hello 表示将工程文件保存到主文件夹下的 hello 目录里。设置结果如图 20-14 所示。

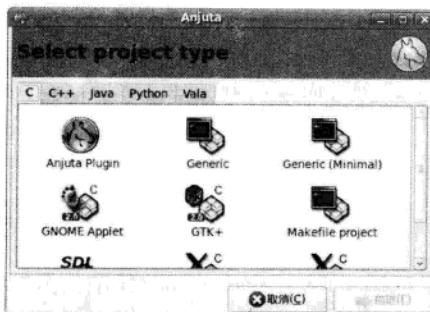


图 20-13 选择工程的类型

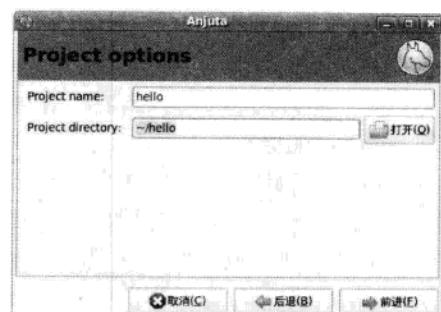


图 20-14 设置工程的属性

(5) 单击“前进”按钮，进入显示新建工程摘要界面，然后直接单击“应用”按钮。Anjuta 开始按照设置创建工程 hello。创建之后，Anjuta 会在文件面板上显示工程 hello 的相关文件，双击由 Anjuta 默认创建 hello.c 文件，代码显示结果如图 20-15 所示。可发现 Anjuta 已写了 helloWorld 的程序。同时，Anjuta 也创建了工程 hello 的配置文件 makefile。

(6) 编译源码文件。打开已编辑好的源文件 hello.c，然后选择“生成”|“编译 (hello.c)”菜单命令，Anjuta 调用编译程序 GCC 编译 hello.c 文件，并生成 hello.o 编译文件，编译结果显示在信息面板上如图 20-16 所示。

(7) 生成工程应用程序。将构建整个工程的应用程序，生成可执行的应用程序文件。选择“生成”|“生成工程”菜单命令，Anjuta 开始执行生成可执行的应用程序，结果信息显示在信息面板上，如图 20-17 所示。经过此步骤后，生成了一个名为 hello 的可执行应用程序。

```
1 // Mode: C; indent-tabs-mode: t; c-basic-offset: 4; tab-width: 4 -*- */\n2 \n3 /* main.c - Auto-generated by Anjuta's Makefile project wizard */\n4 \n5 \n6 \n7 #include <stdio.h>\n8 int main()\n9 {\n10     printf("Hello world\n");\n11 }\n12 \n13
```

图 20-15 Anjuta 打开工程

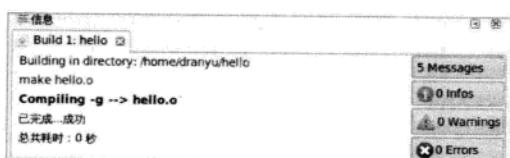


图 20-16 编译文件的结果显示

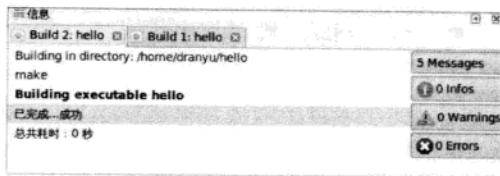


图 20-17 生成可执行应用程序

(8) 在 Anjuta 中直接运行应用程序。Anjuta 集成了终端，编译源码文件后，可使用此终端执行刚生成的可执行文件 hello。选择“Run”|“Program Parameters”菜单命令，弹出 Program parameters 对话框中设置应用程序运行的参数。在 Program 输入栏中输入所需要执行的应用程序名称，如 hello。在“参数”输入栏上输入应用程序的参数，此处程序没有设置参数，所以不需要填写。在“工作目录”下拉列表中选择“运行程序”的命令，默认为工程目录，此为 /home/dranyu/hello。选择“在终端运行”选项，表示应用程序在终端上运行。设置运行应用程序的环境如图 20-18 所示，然后单击“应用”按钮，保存设置。设置完毕后，选择“Run”|“Excute”菜单命令，执行生成的 hello 应用程序，执行结果显示在终端面板上，如图 20-19 所示。



图 20-18 设置运行参数

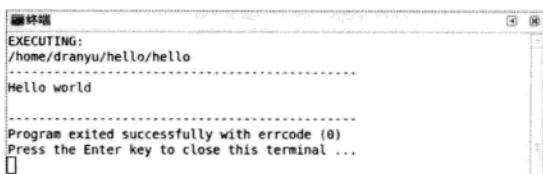


图 20-19 执行程序的结果

20.3.2 Eclipse

Eclipse 是著名的跨平台的自由集成开发环境 (IDE)。最初主要用于 Java 语言开发，但是目前也有人通过插件使其作为其他编程语言，如 C++ 和 Python 的开发工具。针对 C/C++ 编程语言，Eclipse 基金会创建了 Eclipse CDT 项目。Eclipse CDT 是非常强大的编辑器、方便的重构功能、代码历史回溯、版本控制、强大易用的 GCC 工具链前端等。

1. Eclipse CDT 的功能

C 和 C++ 语言都是世界上最流行且被使用最普遍的编程语言。使用框架式设计的 Eclipse 提供了对 C/C++ 的支持。Eclipse CDT 是 Eclipse 的控件，它实现可视化的编程环境，其主要功能如下所示。

- 语法突出显示：CDT IDE 识别 C/C++ 语法，并为语法突出显示提供了完全可配置的代码着色，以及代码格式。
- 提纲：Outline 窗口模块提供了有关出现在源代码中的过程、变量、声明，以及函数的快速视图。利用 Outline，开发者可以方便地找到源代码中的适当引用，甚至搜索所有项目源代码。



- 代码辅助：编辑程序代码时，Anjuta 将在代码模板中查找与程序员的输入相匹配的代码，并在匹配代码列表中显示匹配代码。程序员输入关键字的过程中，Anjuta 会显示关键字的选择框供程序员选择正确的关键字，防止输入错误。
- 代码模板：代码辅助功能使用的代码模板是由标准 C/C++ 语法规则定义的。同时，程序员也可以自定义代码模板来扩展自己的快捷键。
- 代码历史记录：使用这一功能，程序员可跟踪项目源代码在本地的更改。

2. 构建 C++ 开发平台

在下载和安装 CDT 之前，必须确保 GNU C 编译器（GNU C compiler，GCC），以及所有附带的工具（make、binutils 和 GDB）都是可用的。在 Ubuntu 中，C/C++ 调试器 GDB、C/C++ 编译器 GCC 和 make 工具已默认安装。那么，在 Ubuntu 的 Linux 发行版本中安装 Eclipse CDT 的开发环境只需完成以下步骤：

- (1) 安装主框架 Eclipse。在终端输入如下代码添加 Eclipse 应用程序。

```
root@dranyu-desktop:~# apt-get install eclipse
```

(2) 安装语言包 Eclipse-nls。默认安装的 Eclipse 没有中文支持，需要安装语言包 Eclipse-nls 上，执行命令和结果显示如下所示。

```
root@dranyu-desktop:~# apt-get install eclipse-nls
```

- (3) 安装 Eclipse CDT。安装 CDT 的命令执行和结果如下所示。

```
root@dranyu-desktop:~# apt-get install eclipse-cdt
```

安装完毕后，在 Ubuntu 的主菜单上自动添加了“应用程序”|“编程”|Eclipse 启动菜单。

3. Eclipse 的基本操作

选择“应用程序”|“编程”|Eclipse 菜单命令，启动 Eclipse 应用程序，其主界面如图 20-20 所示。

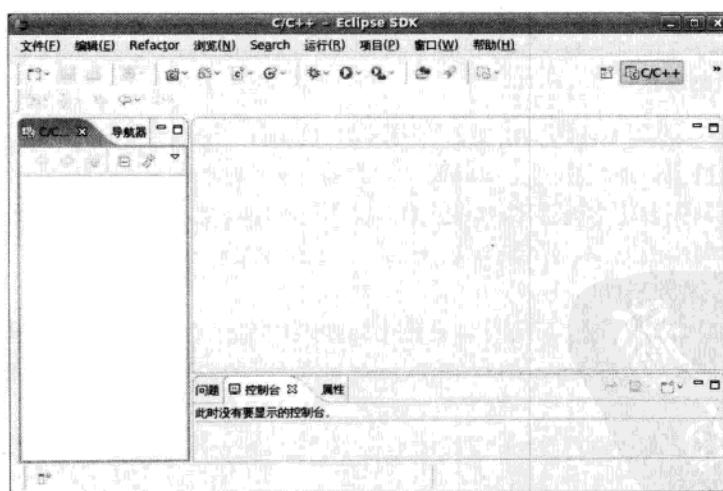


图 20-20 Eclipse 主界面

例 20-4：创建、编辑、编译和执行名为 hello 的 C/C++ 工程项目

使用 Eclipse 创建一个名为 hello 的 C/C++ 工程项目，并编写源代码和编译工程项目，最

后执行生成的程序，具体步骤如下：

(1) 在 Eclipse 主界面上，选择“文件”|“新建”|“项目”菜单命令，弹出“新建项目”对话框，在过滤文本输入栏中输入过滤文本，如 C，查找与 C 匹配的项目。过滤的结果如图 20-21 所示。

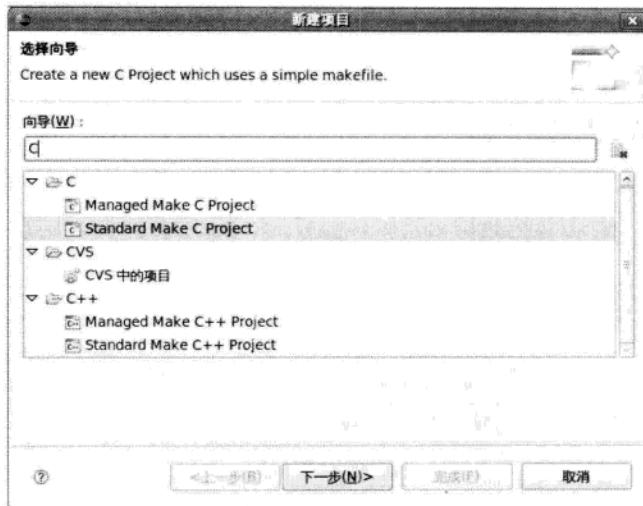


图 20-21 选择需要创建的 C/C++项目

(2) 在“新建项目”对话框中的项目列表下选择“Standard Make Project”选项，然后单击“下一步”按钮，进入设置新项目参数界面，在“项目名”输入栏上输入新建项目的名称，如 hello。其余保持默认设置，如图 20-22 所示。



图 20-22 设置项目名

(3) 下一步的设置是项目的相关设置，如类库的设置、错误处理的设置和环境变量的设置等。保持默认设置，单击“完成”按钮，创建名为 hello 的 C 项目。

(4) 选择“文件”|“新建”|Source file 菜单命令，为 hello 项目创建一个名为 hello.c 的



源文件，在 hello.c 源文件中输入 hello World 的程序代码如图 20-23 所示。

```
#include <stdio.h>
int main(){
    printf("Hello World!\n");
    return 0;
}
```

图 20-23 编写 hello.c

(5) 为 hello 项目添加 build 的配置。用鼠标右键单击界面右边的 hello 项目树上的 hello.c 文件，选择用鼠标右键单击菜单中的“Create Make Target”菜单命令，在弹出的 Create a new Make Target 对话框中的 Target name 输入栏中输入 Eclipse 标识的 target 名称，在 Make Target 输入栏上输入 makefile 标识的 target 名称，其余参数保持默认值，如图 20-24 所示。单击“Create”按钮完成设置。

(6) 编译 hello 项目。用鼠标右键单击 hello.c 文件，在弹出的快捷菜单中选择“Build Make Target”菜单，弹出 Make Targets 对话框如图 20-25 所示，单击“Build”按钮，完成编译操作。



图 20-24 设置 Make Target

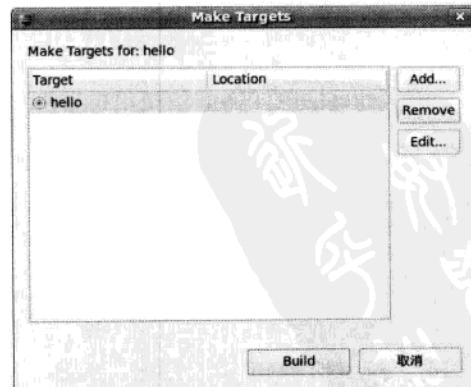


图 20-25 编译 hello 项目

(7) 运行 C/C++ 项目程序。在项目树上, 选中项目文件夹 hello, 然后用鼠标右键单击, 在弹出的快捷菜单中选择“运行” | “运行方式” | “Run Local C/C++ application” 菜单命令, Eclipse 会根据 make target 编译 hello 应用程序, 运行后, 在控制台面板上显示了运行结果, 如图 20-26 所示。

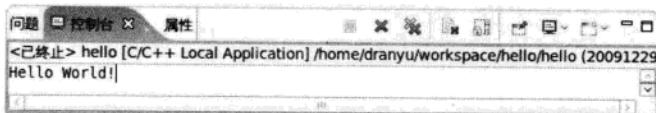


图 20-26 显示执行结果

20.4 C/C++ 程序开发工具



无论是 Anjuta 还是 Eclipse, 它们在编译和调试时都会使用到 C/C++ 编译工具、调试工具和配置文件生成工具。Ubuntu 默认安装了 GCC 编译器、gdb 调试工具和 make 配置文件生成工具, 也就是说 Ubuntu 从安装完毕就具备了编译 C/C++ 程序的能力。

20.4.1 编译器工具 GCC

GCC 是 GNU 推出的功能强大、性能优越的多平台编译器, 是第一款可移植、可优化、支持 ANSI C 的开源 C 编译器。

1. GCC 的文件格式

使用 GCC 编译器过程中会遇到多种文件格式如表 20-7 所示。这些不同格式的文件在 GCC 执行的不同阶段产生或作为 GCC 的文件格式输入。

表 20-7 GCC 识别的文件扩展名

文件扩展名	文件类型
.c	C 语言代码
.C、.cc	C++ 语言代码
.i	预处理后的 C 语言代码
.ii	预处理后的 C++ 语言代码文件
.h	程序所包含的头文件
.s、.S	汇编语言代码
.o	目标代码
.a	静态链接库(程序编译时使用)
.so	动态链接库(程序编译时使用)

2. GCC 的执行过程

GCC 将 C 语言源代码文件生成可执行文件的过程, 不仅仅是一个编译的过程, 它的执行



需要经历 4 个相互关联的步骤：预处理（也称预编译，Pre-Processing）、编译（Compiling）、汇编（Assembling）和链接（linking）。下面就以一个实例来具体查看一下 GCC 是如何完成这 4 个步骤的，如例 20-5 所示。

例 20-5：GCC 的执行过程解析

假设文件~/Document/program/hello.c 的源代码如下：

```
#include<stdio.h>
int main(void)
{
    printf("Hello world!\n");
    return 0;
}
```

(1) 预处理阶段

在此阶段，编译器将上述代码中的 stdio.h 编译进来，并且程序员可以使用 GCC 的选项“-E”查看显示信息，该选项的作用是让 GCC 在预处理结束后停止编译过程，命令如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ gcc -E hello.c -o hello.i
```

在此处，选项“-o”指定生成的目标文件，“i”文件为已经过预处理的 C 语言原始程序。经过预处理后 stdio.h 文件的内容插入到 hello.i 文件中。

(2) 编译阶段

接下来进行的是编译阶段，在此阶段中，GCC 首先要检查代码的规范性、是否有语法错误等，以确定代码实际要做的工作，在检查无误后，GCC 把代码翻译成汇编语言。程序员可以使用“-S”选项查看信息，该选项只进行编译而不进行汇编，生成汇编代码。

```
dranyu@dranyu-desktop:~/Documents/program$ gcc -S hello.i -o hello.s
```

(3) 汇编阶段

汇编阶段是把编译阶段生成的“.s”文件转成目标文件，程序员在此可使用选项“-c”查看汇编代码已转化为“.o”的二进制目标代码，如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ gcc -c hello.s -o hello.o
```

(4) 链接阶段

在成功编译之后，进入链接阶段。完成了链接后，GCC 将目标代码生成可执行文件，完成链接和生成可执行命令如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ gcc hello.o -o hello
```

运行该可执行文件，出现正确的结果如下：

```
dranyu@dranyu-desktop:~/Documents/program$ ./hello
```

```
Hello world!
```

3. GCC 的常用选项

GCC 功能强大，有多达上千个选项，其用户手册也有将近一万行，但是其中的多数选项平时很少用到。其基本语法如下：

GCC [选项] [选项 参数]

gcc 常用选项含义如下所示。

- -c：此选项表示将源代码编译为目标文件，而不连接库。
- -S：此选项表示将所操作的代码文件编译为汇编代码文件。
- -E：此选项表示对源代码进行预处理。预处理之后的代码将送往标准输出。

- **-W warn...**：此选项设置警告，`warn` 是一个参数表示警告的类型。用 `-W all` 开启所有的警告。
- **-I dir...**：此选项表示把 `dir` 加到头文件的搜索路径中，而且 GCC 会在搜索标准头文件之前先搜索 `dir`。`dir` 参数指定搜索的文件目录。如需要设置多个搜索目录，每个目录之间必需使用空格隔开，如 `-I /usr/lib /usr/local/lib`。
- **-l library**：此选项指定在 Gcc 联接过程时应该搜索 `library` 库。参数 `library` 是函数库的路径。
- **-L dir...**：此选项把目录 `dir` 加到库文件的搜索路径中，而且 GCC 会在搜索标准库文件之前先搜索目录 `dir`。
- **-pthread**：`pthread` 库加入对多线程的支持，这为预处理和联接设置了标志。`pthread` 是 POSIX 指定的标准线程库。
- **-std=standard**：此选项用于设置采用的标准。此选项是针对 C 语言而设定的，如 `-std=c99` 表示编译器遵循 c99 标准。此选项较少使用。
- **-o outfile**：此选项指定输出名为 `outfile` 的文件。在不指定时，默认输出文件为 `a.out`。
- **-m machine-option...**：此选项指定所用的操作系统平台。

举例说明 GCC 的使用如下。

例 20-6：将一个源文件编译为一个指定的文件

当程序在一个文件时，开发者可使用 GCC 命令对它进行编译，并输出指定的文件，命令如下：

```
dranyu@dranyu-desktop:~/Documents/program$gcc hello.c -o hello
```

上句表示将 `hello.c` 源文件编译为目标文件 `hello`。

GCC 采用搜索目录的办法来查找所需要的文件，`-I` 选项可以向 GCC 的头文件搜索路径中添加新的目录。

例 20-7：设置头文件

如果在 `/home/dranyu/include` 目录下有编译时所需要的头文件，为了让 GCC 能够顺利地找到它们，就可以使用 `-I` 选项：

```
# gcc hello.c -I /home/dranyu/include -o hello
```

如果使用了不在标准位置的库文件，那么可以通过 `-L` 选项向 GCC 的库文件搜索路径中添加新的目录。

例 20-8：设置链接库文件

如果在 `/home/xiaowp/lib` 目录下有链接时，所需要的库文件 `libhello.so`，为了让 GCC 能够顺利地找到它，可以使用下面的命令：

```
# gcc hello.c -L /home/dranyu/lib -o hello
```

20.4.2 调试工具 gdb

GNU 的调试器称为 `gdb`，该程序是一个交互式工具，工作在字符模式。`gdb` 是功能强大的调试程序，可完成如下的调试任务。

- 设置断点。
- 监视程序变量的值。



- 程序的单步执行。
- 修改变量的值。

运行 `gdb` 调试程序时通常使用的命令格式如下所示：

```
gdb programe
```

其中 `program` 是需要调试的应用程序的文件名。

例 20-9：调试 C/C++ 程序

`hello` 程序是例 20-5 中源文件 `hello.c` 经过 `GCC` 编译后的可执行程序，其保存在 `~/Documents/program` 目录下。调试 `hello` 程序的命令如下：

```
dranyu@dranyu-desktop:~/Documents/program$ gdb hello
```

`hello` 为应用程序的名称。在终端上执行命令和结果显示如图 20-27。

The screenshot shows a terminal window titled "dranyu@dranyu-desktop: ~/Documents/program". The command "gdb hello" has been entered, and the output shows the GNU GDB version 6.8-debian, the copyright notice from 2008, and the GPL license information. The prompt "(gdb)" is visible at the bottom.

图 20-27 启动 `gdb`

启动 `gdb` 调试工具后，进入 `gdb` 的调试界面。在 `gdb` 提示符处键入 `help`，将列出各命令的分类，主要的分类包括如下。

- `aliases`: 命令别名。
- `breakpoints`: 断点定义。
- `data`: 数据查看。
- `files`: 指定并查看文件。
- `internals`: 维护命令。
- `running`: 程序执行。
- `stack`: 调用栈查看。
- `status`: 状态查看。
- `tracepoints`: 跟踪程序执行。

键入 `help` 后跟命令的分类名，如 `help aliases`，可获得该类命令的详细清单。

在使用 `gdb` 调试工具的时候，常用的 `gdb` 命令如表 20-8 所示。

表 20-8 `gdb` 常用命令

命 令	说 明
<code>file</code>	此命令表示装入想要调试的可执行文件。如 <code>file hello</code> 表示装入 <code>hello</code> 可执行文件
<code>kill</code>	此命令表示终止正在调试的程序
<code>list</code>	此命令表示列出产生执行文件的源代码的一部分
<code>next</code>	此命令表示执行一行源代码，但不进入函数内部

续表

命 令	说 明
step	此命令表示执行一行源代码而且进入函数内部
run	此命令表示执行当前被调试的程序
quit	此命令表示终止并退出 gdb 程序
watch	此命令表示使程序员能监视一个变量的值而不管它何时被改变
make	此命令表示使程序员不退出 gdb 就可以重新生成可执行文件
shell	此命令表示使程序员能不离开 gdb 界面，就可执行 UNIX shell 命令
print 变量名	此命令表示显示变量名保存的值
break NUM	此命令表示在第 NUM 行中设置断点
bt	此命令表示显示所有的调用栈帧。该命令可用来显示函数的调用顺序

例 20-10：装入可执行文件

在 gdb 调试工具 gdb 提示符界面上装入一个可执行的文件。

```
(gdb) file hello
```

其中 hello 是一个可执行的文件。此语句表示装入 hello 可执行文件，并对 hello 文件进行调试。

例 20-11：设置断点

在装入的 hello 程序中设置断点，其命令如下所示：

```
(gdb) break 11
```

其中 11 就是源代码中的第 11 行。此语句表示在源代码中的第 11 行设置一个伙伴。

例 20-12：显示程序中变量的值

查看调试程序中设置的变量，命令如下所示：

```
(gdb) print hello
```

hello 是所调试程序中的一个变量，使用 print 命令将 hello 变量的内容显示出来。

20.4.3 make 工具

在应用程序的开发过程中，若每次修改源文件都需要重新输入编译命令。这种做法很繁琐而且开发效率不高。为了能快速组织源文件进行编译，Ubuntu Linux 中提供了 make 工具。make 的运行过程前，必须编写 Makefile 文件。Makefile 文件描述了软件包中文件之间的关系，提供更新每个文件的命令。make 实际上是根据 Makefile 来进行编译的。

当使用 make 工具进行编译时，工程中以下几种文件将会被编译：

(1) 所有没有被编译过的源文件将在此次执行 make 时被重新编译

每一个在上次执行 make 之后修改过的 C 源代码文件在本次执行 make 时将会被重新编译

(2) 头文件在上一次执行 make 之后被修改，则所有包含此头文件的 C 源文件在本次执行 make 时将会被重新编译。

Makefile 是一个文本形式的数据库文件，其中包含一些规则来告诉 make 处理哪些文件，以及如何处理源码文件。一个简单的 Makefile 描述规则组成：

```
target: prerequisites
```



command

其中 target、prerequisites 和 command 的含义如下所示。

- **target:** 规则的目标。通常是最需要生成的文件名或者为了实现这个目的而必需的中间过程文件名。可以是.o 文件，也可以是最后的可执行程序的文件名等。另外，目标也可以是一个 make 执行的动作的名称，如目标“clean”，我们称这样的目标是“伪目标”。
- **prerequisites:** 规则的依赖。生成规则目标所需要的文件名列表。通常一个目标依赖于一个或者多个文件。
- **command:** 规则的命令行。是规则所要执行的动作(任意的 shell 命令或者是可在 shell 下执行的程序)。它限定了 make 执行这条规则时所需要的动作。

例 20-13：创建 Makefile 文件

为 hello 应用程序创建一份 Makefile 文件，使用 hello.c 作为源文件，hello 为应用程序的输出名称，此 Makefile 的写法如图 20-28 所示。

```
1 hello:hello.o
2     gcc -o hello hello.o
3 hello.o:hello.c
4     gcc -o hello.o -c hello.c
5 clean:
6     rm -f *.o hello
```

图 20-28 Makefile 的编写

第 1、2 行组成了一个 Makefile 规则，其中 hello 是 target，hello.o 是生成 hello 所需使用的源文件，第 2 行是 command 命令行。gcc -o hello hello.c 表示执行 gcc 编译程序完成将 hello.o 编译成 hello 可执行文件的过程。

第 3、4 行也组成了一个 Makefile 规则，hello.o 是此规则的 target，hello.c 是生成 hello.o 所需要的源文件。第 4 行是命令行，将 hello.c 源文件编译为 hello.o，为 hello 的生成提供 hello.o 文件。

第 5、6 行组成了一个 target 为 clean 的规则。clean 为此规则的目标。第 6 行是命令行，其表示删除在编译过程中生成的目标代码文件 (*.o) 和可执行文件 hello。

补充说明

在编写 Makefile 规则的命令行时，需要注意的是命令行的开头必须使用 Tab 键开头。这是初学者最容易忽略的。

make 工具的使用很简单，其使用方法为，在终端上进入到 Makefile 所在的文件目录中，然后执行 make 命令，这样 make 工具就会搜索当前目录上的 Makefile 文件，然后按照 Makefile 上的规则执行相应的命令。make 工具命令的使用方法如下：

```
make [target]
```

其中 target 表示在 Makefile 文件中定义的 target。在不使用参数的情况下，系统默认 target 指向 Makefile 文件中第一个命令。

执行 make 命令生成 hello 应用程序和使用 make 删除由 make 生成的所有文件，执行命令和结果显示如图 20-29 所示。

```

dranyu@dranyu-desktop:~/workspace/hello$ ls
hello.c Makefile
dranyu@dranyu-desktop:~/workspace/hello$ make
gcc -o hello.o -c hello.c
gcc -o hello hello.o
dranyu@dranyu-desktop:~/workspace/hello$ ls -l
总用量 24
-rwxr-xr-x 1 dranyu dranyu 9145 2009-12-29 09:40 hello
-rw-r--r-- 1 dranyu dranyu 70 2009-12-29 08:54 hello.c
-rw-r--r-- 1 dranyu dranyu 864 2009-12-29 09:40 hello.o
-rw-r--r-- 1 dranyu dranyu 103 2009-12-29 09:14 Makefile
dranyu@dranyu-desktop:~/workspace/hello$ make clean
rm -f *.o hello
dranyu@dranyu-desktop:~/workspace/hello$ ls -l
总用量 8
-rw-r--r-- 1 dranyu dranyu 70 2009-12-29 08:54 hello.c
-rw-r--r-- 1 dranyu dranyu 103 2009-12-29 09:14 Makefile
dranyu@dranyu-desktop:~/workspace/hello$ 

```

图 20-29 make 命令

在上图中，可发现执行不带参数的 make 执行了 Makefile 中的前两个命令，完成了 hello 和 hello.o 的生成过程。因为在执行第一个命令时需要使用 hello.o 源文件，而 hello.o 文件的生成是由第二个命令完成的，所以 make 执行了前两个命令。执行 make clean 命令时，make 调用了 Makefile 文件中 target 为 clean 命令行中的命令 rm -f *.o hello。由此可知，make 命令的执行总是调用 Makefile 中设置的命令。



20.5 小结

本章主要讲述了 Ubuntu 下的 C/C++ 应用程序的编程。讲述了 vim 和 Emacs 两个在 Linux 下常用的代码编辑软件的安装和应用。Anjuta 集成开发环境是广泛应用的 C/C++ 应用程序开发环境。本章以一个 hello world 的项目讲解了 Anjuta IDE 的基本应用，包括创建新项目、编译项目和调试项目等。Eclipse CDT 是一个广受 C/C++ 开发人员欢迎的应用程序。本章介绍了 Eclipse 的安装和将 Eclipse 配置为 C/C++ 的开发环境。最后讲述了 Linux 下 C/C++ 的编译器 GCC、调试器 gdb 和工程链接工具 make 的应用。



20.6 习题

1. 打开终端，在终端上使用 apt-get 命令安装 vim 软件包。
2. 完成上题的操作后，在终端使用 vim 命令，创建一个新的文本文件 readme.txt。然后输入“这是使用 vim 应用程序编辑的一个示范文本。”，保存文件并退出 vim 应用程序。
3. 在终端使用 apt-get 命令安装 Emacs32 软件包。



4. 完成上题操作后，使用 Emacs 应用程序编辑一个 hello world 的 C 语言程序（参考例 20-5），然后使用 GCC 命令编译此 C 语言程序，查看执行结果。
5. 在 Ubuntu 的新立得软件包管理器中安装 Anjuta 编程环境。
6. 安装了 Anjuta 软件后，按照例 20-3 的步骤创建一个名为 hello 的 C 项目。编辑 hello.c 文件，使得程序实现输出“Hello, Anjuta!”字符串的功能。然后对 hello 项目进行编译、运行和调试。
7. 打开新立得软件包，安装 Eclipse 软件包和 Eclipse CDT 软件包，构建 C/C++ 编辑环境。
8. 创建桌面的 Eclipse 的启动器，然后启动 Eclipse 应用程序，创建一个名为 hello 的 C 项目。为 hello 项目添加一个源文件命名为 hello.c。编写 hello.c，使其实现输出“Hello, Eclipse！”字符串的功能，然后编译此项目和运行此项目。（提示：参考例 20-4 Eclipse 的基本操作）
9. 使用 GCC 编译一个已存在的 hello.c 文件，将 hello 作为生成的目标文件。
10. 使用 gdb 工具调试由上题生成的 hello 可执行文件。
11. 为 hello.c 编写一个 Makefile，然后执行 make 命令编译 hello.c 源文件。hello.c 的内容如例 20-5 所示。Makefile 文件的编写如例 20-13 所示。

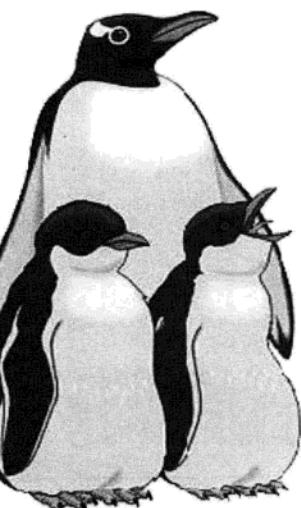


Linux

第 21 章

shell 编程

作为解释程序，shell 接收在终端上输入的每一条命令，然后解释此命令并执行它。除此之外，shell 还是一种程序设计语言。由 shell 所编写的脚本是解释型的，而不是编译型的。shell 脚本中是 shell 解释程序下执行的，shell 解释程序从脚本中读取每一行命令并执行它。Ubuntu 系统默认使用的 shell 是 Bash shell。第 10 章已讲解 shell 作为解释程序的内容，本章就不再讲述。本章主要介绍 Bash shell 程序编写语法，以及如何编写 Bash shell 脚本。





21.1 Bash 简介

Bash 是兼容 sh shell 且从 Korn shell (ksh) 和 C shell (csh) 中整合了一些有用特性。它遵循 IEEE POSIX P1003.2/ISO 9945.2 Shell 和工具标准，提供了基于 sh 的编程和交互的功能改进，其中包括命令行编辑、无限制的历史命令、作业控制、shell 函数和别名、无大小限制的索引数组和以 2 到 64 为基础的整数算法等。Bash 的兼容性良好，可在不修改脚本的情况下运行其他 shell 编写的 shell 脚本。

和其他的 GNU 项目一样，Bash 遵循自由软件的规范，即其使用者有使用、复制、散布、研究、改写、再利用该软件的自由。这使得 Bash 成为许多程序员的首选，从而有许多程序员为其添加了许多其他 shell 无法提供的额外特性。

21.1.1 Bash 的优势

Bash 是 Ubuntu Linux 系统默认使用的 shell，它由 Brian Fox 和 Chet Ramey 共同编写完成，是 Bourne Again Shell 的缩写，其拥有内部命令 40 个。目前，多数 Linux 版本使用它作为默认的 shell，是因为它有诸如以下的特色：

- (1) 支持类似 DOS 里的 doskey 的功能，用方向键可查阅和快速输入并修改命令。
- (2) 提供自动补全功能。
- (3) 拥有完整的帮助文档。

21.1.2 Bash 脚本的编写与运行

Bash 脚本的编写可使用任意的文本编辑器，如 VIM、Emacs 和 Gedit 等。Bash 脚本的编写和运行需要经过如下步骤：

- (1) 使用文本编辑器编写 Bash 脚本并保存脚本。
- (2) 修改脚本的访问属性。
- (3) 运行 Bash 脚本。

以一个实例讲述 Bash 脚本的编写与运行过程，如例 21-1 所示。

例 21-1：编写一个输出信息的 Bash 脚本

- (1) 使用 Gedit 创建一个名为 test 的空文件，在 test 文件中输入如下代码：

```
#!/bin/bash //Bash 脚本的标准开始
#This script is a test!
echo -n "Date and time is : " //在终端显示“Date and time is :”字符串
date //调用 date 命令，显示系统日期和时间
echo -n "Your Current directory is : " //在终端显示“Your Current directory is :”
pwd //调用 pwd 命令，显示当前的工作目录
#end //注释行，表示 Bash 脚本的结束
```

- (2) 查看 test 的访问权限并修改其访问权限，执行命令和结果显示如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ ls -l
总用量 4
-rw-r--r-- 1 dranyu dranyu 119 2010-01-03 16:46 test
```

```
dranyu@dranyu-desktop:~/Documents/program$ chmod 755 test //设置 test 可执行权限
dranyu@dranyu-desktop:~/Documents/program$ ls -l
总用量 4
-rwxr-xr-x 1 dranyu dranyu 119 2010-01-03 16:46 test
```

(3) 运行 Bash 脚本。若当前工作目录是 test 脚本所在目录，可执行./test 命令运行 test 脚本如下所示。否则输入 test 的绝对路径。

```
dranyu@dranyu-desktop:~/Documents/program$ ./test //执行 test 脚本
Date and time is : 2010 年 01 月 03 日 星期日 16:48:16 CST
Your Current directory is :/home/dranyu/Documents/program
```

补充说明

“#!/bin/bash”是 Bash 脚本的第一句话，指明运行脚本文件所用的 shell 路径。一般来说，创建一个新的文件时系统将文件设置为不可执行。shell 脚本只有在用户拥有执行权限时才可被执行。使用 chmod 命令修改 shell 的访问权限也就成为 shell 脚本运行的一个重要的步骤。



21.2 Bash 的语法

Bash 语法是编写 Bash 的规则。只有了解 Bash 语法，程序员才能准确无误地编写 Bash 脚本。Bash 的语法涉及 Shell 变量、函数、结构语句和 shell 内置命令等内容。

21.2.1 shell 变量

变量是脚本编程中进行数据保存的一种方法。变量不过是计算机为了保留数据项，而在内存中分配的一个位置或一组位置的标识或名字。变量既可以出现在算术操作中，也可以出现在字符串分析过程中。

shell 中有 3 种变量：系统变量、环境变量和用户变量。其中用户变量用得最多，用于保存用户定义的值；系统变量对参数判断和命令返回值判断的时候使用；环境变量则多是用于程序运行的时候。

1. 系统变量

在 Bash 中，系统变量又可分为位置变量和内部变量。

- 位置变量

位置参数是标识当前脚本运行时输入的参数位置，格式为\$N。其中\$表示此字符串为 shell 的变量，N 是自然数值，如 0、1 等。

例 21-2：位置变量

(1) 编写一个名为 test 的 bash 脚本，其代码如下所示：

```
#!/bin/bash
echo "程序名：" $0
echo "第 1 个参数的值：" $1
echo "第 2 个参数的值：" $2
#end
```



(2) 修改 test 的访问权限，然后执行命令和显示结果如下：

```
dranyu@dranyu-desktop:~/Documents/program$ chmod 755 test //修改访问权限  
dranyu@dranyu-desktop:~/Documents/program$ ./test parameter1 parameter2 //执行脚本  
程序名：test  
第1个参数的值：parameter1  
第2个参数的值：parameter2
```

从显示结果中可知：\$0 表示是当前所执行的脚本名称；\$1 表示位于脚本命令后的第一个参数；\$2 表示位于脚本命令后的第二个参数。以此类推，除 N 为 0 外，\$N 表示位于脚本命令后的第 N 个参数。

● 内部变量

在 Bash shell 中，有一部分的变量是由 shell 自身定义的，用户只能使用这些变量而不能重新设置它们。这些变量如表 21-1 所示。

表 21-1 Bash 的内部变量

内部变量	说 明
\$#	命令行上的参数个数
\$?	前一个命令或函数的返回值
\$* \$@	命令行所有参数构成的一个字符串，其格式为“参数1 参数2...”
\$\$	当前运行进程的进程 ID
\$!	最后一个后台运行进程的进程

例 21-3：显示一个脚本中内部变量的值

编写一个名为 test 的 Bash 脚本，其代码如下所示：

```
#!/bin/bash  
echo "程序名：" $0  
echo "参数个数：" $#  
echo "前一个命令的返回值：" $?  
echo "显示所有参数：" $*  
echo "显示所有参数：" $@  
echo "此进程的进程 ID：" $$  
echo "最后一个后台运行进程的进程：" $!  
#end
```

修改 test 的访问权限，然后执行命令和显示结果如下：

```
dranyu@dranyu-desktop:~/Documents/program$ chmod 755 test //修改访问权限  
dranyu@dranyu-desktop:~/Documents/program$ vim & //在后台执行 vim 程序  
[1] 7689  
dranyu@dranyu-desktop:~/Documents/program$ ./test para1 para2 para3 //执行脚本  
程序名：./test  
参数个数：3  
前一个命令的返回值：0  
显示所有参数：para1 para2 para3  
显示所有参数（使用双引号分隔）：para1 para2 para3  
此进程的进程 ID：7690  
最后一个后台运行进程的进程：  
[1]+ Stopped vim
```

2. 用户变量

用户自定义的变量必须由字母数字和下画线组成，并且第一个字符不能为数字，区分大

小写。shell 的用户变量不需要专门的声明语句，通过赋值语句即可完成变量的声明及赋值。

- 变量赋值

对于用户变量，赋值语句的语法格式如下所示：

```
Name=string
```

其中 Name 为变量名，string 是赋给变量的字符串值，而符号“=”是赋值号。shell 语言是非类型的解释型语言，给一个变量赋值实际上就是定义了变量，而且可以赋不同类型的值。

注意

在定义变量的时候，变量与“=”之间不能有空格，同样“=”与变量值之间也不能有空格。

- 引用变量

在引用变量时需要加“\$”符号。变量之间也可以相互赋值。引用变量有两种方式：`$varname` 和 `${varname}`。为防止变量在字符串中产生歧义建议使用第二种方式，引用未定义的变量其值为空。

例 21-4：自定义变量的赋值和引用

(1) 使用 Gedit 文本编辑器，编写一个名为 test，输入如下代码：

```
#!/bin/bash
userName="dranyu"; // 定义 userName 变量，并赋值为 dranyu
echo "Hello,$userName" // 使用$userName 格式引用 userName 变量
echo "Hello,${userName}" // 使用${userName}格式引用 userName 变量
echo "Hello,$dranyu" // 引用定义的 dranyu 变量
echo "${userName}" // 使用双引号
echo '${userName}' // 使用单引号
```

(2) 修改 test 的访问权限，然后执行命令和显示结果如下：

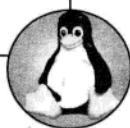
```
dranyu@dranyu-desktop:~/Documents/program$ ./test // 执行 test 脚本
Hello,dranyu
Hello,dranyu
Hello,
dranyu
${userName}
```

由上述例可知，使用双引号与使用单引号将变量的引用括起来后，输入的结果不同。使用单引号后，命令不会将\$userName 解释为 dranyu，即此时\$userName 只是一个普通的字符串。

3. shell 环境变量

shell 环境变量就是所有的 shell 程序都会接受的参数，shell 程序启动时，都会接收一组变量，这组变量就是环境变量。常用的环境变量包括以下。

- PATH：指定命令搜索路径，以冒号为分隔符，注意与 DOS 下不同的是，当前目录不在系统路径里。
- HOME：用户 home 目录的路径名，是 cd 命令的默认参数。
- COLUMNS：指定命令行编辑模式下可使用的命令行的长度。
- EDITOR：指定默认的行编辑器。
- VISUAL：指定默认的可视编辑器。



- FCEDIT: 指定命令 fc 使用的编辑器。
- LANG: 指定系统的语言。
- IFS: 定义 shell 使用的分隔符。
- LOGNAME: 保存当前登录的用户名。
- MAIL: 指向一个需要 shell 监视其修改时间的文件。当该文件修改后, shell 将发消息 “You have mail” 给用户。
- MAILCHECK: 指定 shell 检查 MAIL 文件的周期, 单位是秒。
- MAILPATH: 功能与 MAIL 类似, 但可以是一组文件, 文件名之间以冒号分隔。
- SHELL: 指定当前使用 shell 的路径名。
- TERM: 指定模拟终端的类型。
- TMOUT: 指定 shell 自动退出的时间。其值若为 0 则禁止 shell 自动退出。
- PS1: 指定在终端上主命令的提示符。
- PS2: 指定二级命令提示符。
- PS3: 指定 select 命令的提示符。
- PS4: 指定调试命令的提示符。
- MANPATH: 指定寻找手册页的路径。使用多个路径时, 各路径之间以冒号分隔。
- LD_LIBRARY_PATH: 指定寻找库的路径。使用多个路径时, 各路径之间以冒号分隔。

若希望自定义的变量让其他所有的 shell 程序使用, 即定义新的环境变量, 则可使用 export 关键字将自定义变量导入到 shell 的配置文件中, 如例 21-5 所示。

例 21-5: 设置环境变量的值

```
dranyu@dranyu-desktop:~/Documents/program$ export PATH=/home/swliao/bin:$PATH //往  
环境变量 PATH 中添加路径/home/swliao/bin
```

例 21-6: 显示 shell 环境变量的当前值

(1) 使用 Gedit 文本编辑器创建一个名为 test 的脚本, 往 test 输入如下代码:

```
#!/bin/bash  
echo "当前用户的文件夹: ${HOME}"  
echo "当前使用的 shell 路径: ${SHELL}"  
echo "当前登录的用户名: ${LOGNAME}"  
echo "当前的工作路径: ${PATH}"  
echo "当前使用的系统语言: ${LANG}"
```

(2) 修改 test 的访问权限, 然后执行命令和显示结果如下:

```
dranyu@dranyu-desktop:~/Documents/program$ ./test  
当前用户的文件夹: /home/dranyu  
当前使用的 shell 路径: /bin/bash  
当前登录的用户名: dranyu  
当前的工作路径: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games  
当前使用的系统语言: zh_CN.UTF-8
```

补充说明

使用 env 命令可查看当前登录到 shell 终端的环境变量。环境变量是会随着不同的用户登录而改变的。

21.2.2 条件语句

曾用过如 C、Pascal、Python 或 Perl 等过程语言编程的程序员一定熟悉“if”条件语句和“for”循环语句的标准结构。在 Ubuntu Linux 下，Bash 有自己的标准结构版本。其中条件语句的格式如下所示：

```
if 判断表达式 1
then
    命令 1
fi
或
if 判断表达式 1
then
    命令 1
    .....
else
    命令 3
.....
fi
或
if 判断表达式 1
then
    命令 1
    .....
[elif 判断表达式 2
then
    命令 2
....]
else
    命令 3
.....
fi
```

以上 3 种格式是 If 条件语句的 3 种形式：if 命令简单形式、if/else 命令双路决策形式和 if/elif/else 命令多路决策形式。

条件语句涉及判断表达式。在 Bash 脚本中，判断表达式可分为文件判断比较表达式、字符串判断比较表达式和算术判断比较表达式。文件判断比较表达式主要是对文件进行判断比较处理，如表 21-2 所示；字符串判断表达式主要是对字符串进行判断比较处理，如表 21-3 所示；而算术判断比较表达式则是对数值进行比较判断，如表 21-4 所示。

表 21-2 文件比较表达式

运 算 符	描 述	示 例
-e filename	如果 filename 存在，则为真	[-e /var/log/syslog]
-d filename	如果 filename 为目录，则为真	[-d /tmp/mydir]
-f filename	如果 filename 为常规文件，则为真	[-f /usr/bin/grep]
-L filename	如果 filename 为符号链接，则为真	[-L /usr/bin/grep]
-r filename	如果 filename 可读，则为真	[-r /var/log/syslog]
-w filename	如果 filename 可写，则为真	[-w /var/mytmp.txt]



续表

运算符	描述	示例
-x filename	如果 filename 可执行，则为真	[-L /usr/bin/grep]
filename1-nt filename2	如果 filename1 比 filename2 新，则为真	[/tmp/install/etc/services -nt /etc/services]
filename1-ot filename2	如果 filename1 比 filename2 旧，则为真	[/boot/bzImage -ot arch/i386/boot/bzImage]

表 21-3 字符串比较表达式

运算符	描述	示例
-z string	如果 string 长度为零，则为真	[-z "\$myvar"]
-n string	如果 string 长度非零，则为真	[-n "\$myvar"]
string1= string2	如果 string1 与 string2 相同，则为真	["\$myvar" = "one two three"]
string1!= string2	如果 string1 与 string2 不同，则为真	["\$myvar" != "one two three"]

表 21-4 算术比较表达式

运算符	描述	示例
num1-eq num2	num1 等于 num2	[3 -eq \$mynum]
num1-ne num2	num1 不等于 num2	[3 -ne \$mynum]
num1-lt num2	num1 小于 num2	[3 -lt \$mynum]
num1-le num2	num1 小于或等于 num2	[3 -le \$mynum]
num1-gt num2	num1 大于 num2	[3 -gt \$mynum]
num1-ge num2	num1 大于或等于 num2	[3 -ge \$mynum]

例 21-7：条件语句的应用

(1) 为了说明首先需要将 test 脚本所在目录的结构文件内容修改为如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ ls -Rl //使用 ls 命令罗列目录文件
1 .: //“.”表示当前目录
2 总用量 8
3 drwxr-xr-x 2 dranyu dranyu 4096 2010-01-05 08:26 fold1
4 -rwxr-xr-x 1 dranyu dranyu 60 2010-01-05 00:30 test
5
6 ./fold1: //“./fold1”表示.目录下的 fold1 目录
7 总用量 4
8 -rw-r--r-- 1 dranyu dranyu 6 2010-01-05 08:26 file1
```

从第 3、4 行可知与 test 脚本在同一个目录的还有 fold 文件夹；第 8 行显示 fold 文件夹拥有一个名为 file 的文件。

(2) 编写 test 脚本，代码如下：

```
#!/bin/bash
echo "输入格式为：程序名 文件路径 人名字符串 数值1 数值2"
path=$1 // $1 表示输入脚本命令时所带的第一个参数，下面的以此类推
str=$2
num1=$3
num2=$4
#文件的比较
if [ -d $path ] //判断$path 所指引的文件是否为文件夹
```

```

then
echo "${path}是一个文件夹。"
elif [ -f $path ] //判断$path 所指引的文件是否为文件
then
echo "${path}是一个文件。"
else
echo "找不到${path}文件或文件夹。"
fi
#字符串的比较
if [ "${str}" = "dranyu" ] //判断字符串$str 是否与 dranyu 相同
then
echo "早上好, ${str}"
elif [ "${str}" = "sunny" ] //判断字符串是否与 sunny 相同
then
echo "下午好, ${str}"
else
echo "你好, ${str}。我不认识你。"
fi
#数值的比较
if [ ${num1} -eq ${num2} ] //判断$num1 是否等于$num2
then
echo "${num1}等于${num2}"
elif [ ${num1} -lt ${num2} ] //判断$num1 是否小于$num2
then
echo "${num1}小于${num2}"
elif [ ${num1} -le ${num2} ] //判断$num1 是否小于或等于$num2
then
echo "${num1}小于或等于${num2}"
elif [ ${num1} -gt ${num2} ] //判断$num1 是否大于$num2
then
echo "${num1}大于${num2}"
elif [ ${num1} -ge ${num2} ] //判断$num1 是否大于或等于$num2
then
echo "${num1}大于或等于${num2}"
else
echo "${num1}与${num2}不能比较"
fi

```

(3) 运行 test 脚本, 执行命令结果如下所示:

```

dranyu@dranyu-desktop:~/Documents/program$ ./test fold1/file1 dranyu 12 12 //执行 test
命令, 第 1 个参数 fold1/file1, 第 2 个参数 dranyu, 第 3、4 个参数是 12。
输入格式为: 程序名 文件路径 人名字符串 数值 1 数值 2
fold1/file1 是一个文件。
早上好, dranyu
12 等于 12
dranyu@dranyu-desktop:~/Documents/program$ ./test fold1 sunny 21 12 //执行 test 命令,
第 1 个参数 fold1, 第 2 个参数 sunny, 第 3 个参数是 21, 第 4 个参数是 12。
输入格式为: 程序名 文件路径 人名字符串 数值 1 数值 2
fold1 是一个文件夹。
下午好, sunny
21 大于 12
dranyu@dranyu-desktop:~/Documents/programs$ ./test fold sundranyu 11 12 //执行 test 命令,
第 1 个参数 fold, 第 2 个参数 sundranyu, 第 3 个参数是 11, 第 4 个参数是 12。
输入格式为: 程序名 文件路径 人名字符串 数值 1 数值 2
找不到 fold 文件或文件夹。

```



你好，sundranyu。我不认识你。

11 小于 12

补充说明

使用 if 条件语句时，表达式一定要写在中括号里，而表达式与 “[]” 之间一定要保持至少一个空格的位置。如 [“\$str” = “dranyu”]，“[“与”\$str”之间必须要有空格，同样“dranyu”与 “]” 也必须要有空格。对于初学者来说，空格是十分容易被忽视的。

21.2.3 循环语句

在 Bash shell 中，循环语句用于将一个或一组命令执行指定次数，或者一直执行到满足某个条件为止。Bash shell 提供了 3 种形式的循环语句：for 循环语句、while 循环语句和 until 循环语句。

1. for 循环语句

for 循环语句主要用于在某个数据列表上将执行命令指定次数。for 循环语句的语法结构如下：

```
for 变量 in 数据列表
do
    循环体
done
或
for (( 变量; 判断表达式; 增量 ))
do
    循环体
done
```

其中变量指的是 shell 的变量；数据列表是以空格分开的数据；do 和 done 关键字之间是循环体。

补充说明

第二种写法比较简单。程序员都应该知道其与 C 中的 for 循环语句相似。这种写法在 21.4 节中讲述。

例 21-8：for 循环语句应用

(1) 编写 forShell.sh 脚本，代码如下所示：

```
#!/bin/bash
echo "Say hello:"
for userN in dranyu sunny sundranyu tom jim //userN 是变量，in 是关键字，后面是数据列表
do //循环体开始
echo "Hello, ${userN}" //引用循环变量 userN
done //循环体的结束
```

注意

循环变量用来控制 for 循环体的循环。

(2) 修改脚本的访问权限，执行脚本命令获得如下结果：

```
dranyu@dranyu-desktop:~/Documents/program$ ./forshell.sh
Say hello:
```

```
Hello,dranyu
Hello,sunny
Hello,sundranyu
Hello,tom
Hello,jim
```

在上例的 for 循环语句中，userN 是循环变量；“dranyu sunny sundranyu tom jim” 是数据列表。for 的循环体是 for 循环所要执行的动作。for 开始的时候将数据列表中的第一个参数赋值 dranyu 给循环变量 userN，然后执行循环体的代码，这为一趟；然后，for 将下一个参数值 sunny 赋给循环变量 userN，再执行循环体代码，完成第二趟；如此循环直到将数据列表中的数据都走完一趟后结束，执行命令完成后退出 for 循环。

例 21-9：命令参数作为 for 循环的数据列表

(1) 编写 sayHi.sh 脚本，代码如下所示：

```
#!/bin/bash
for userN in $*          //*$是命令的所有参数，与$@同。
do
echo "Hi,${userN}"    //循环体
done
```

(2) 修改脚本的访问权限，执行脚本命令和结果显示如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ ./sayHi.sh dranyu sunny sundranyu
Hi,dranyu
Hi,sunny
Hi,sundranyu
```

2. while 循环语句

while 循环语句的含义是当满足条件的时候执行循环体，即当判断表达式命令的退出状态为 0 时执行循环体，否则退出循环。命令退出状态 0 表示命令的执行是正确的。while 循环语句的语法结构如下所示：

```
while 判断表达式
do
循环体
done
```

当判断表达式的命令执行的结果为真时，执行循环体的命令，直至判断表达式中命令的执行结果为假时退出循环。

例 21-10：使用 while 语句执行循环体 10 次

在 while 循环体显示当前 while 循环的趟数信息，实现的具体步骤如下：

(1) 编写 whileShell.sh 脚本，代码如下所示：

```
#!/bin/bash
num=0
while [ ${num} -lt 10 ]           //判断${num}的值
do
    num=$(( ${num} + 1 ))        //使用 Bash Shell 的算术，使 num 的值加 1
    echo "第${num}次执行循环体。"   //显示信息，${num}表示循环变量 num 的当前值
done
```

(2) 修改脚本的访问权限，执行脚本和结果显示如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ ./whileShell.sh //执行whileShell.sh脚本
第 1 次执行循环体。
第 2 次执行循环体。
第 3 次执行循环体。
```



第 4 次执行循环体。
第 5 次执行循环体。
第 6 次执行循环体。
第 7 次执行循环体。
第 8 次执行循环体。
第 9 次执行循环体。
第 10 次执行循环体。

补充说明

可以使用 shell 结构来执行简单的整数运算。只需将特定的算术表达式用 "\$((" 和 "))" 括起，Bash 就可以计算表达式。如，`num=${{ $num + 1 }}`语句中，\$num + 1 与其他的高级语言编程语法表达一样。这样 shell 就会将\$num 的数值加上 1 然后赋值给 num 变量。

例 21-11：使用 while 语句与用户交互

使用 while 语句测试用户在终端输入的字符串是否与“China”匹配，若匹配就输出“恭喜你，答对了。”的信息，否则输出“答错，请重新输入：”的信息，编写 `whileShell2.sh` 脚本并运行脚本如下：

(1) 编写 `whileShell2.sh` 脚本，代码如下所示：

```
#!/bin/bash
echo "中国的英文怎么写："
read answer //使用 read 命令获取 shell 终端上用户的输入字符串
while [ "${answer}" != "China" ]
do
    echo "答错，请重新输入："
    read answer //再次等待获取用户的输入
done
echo "恭喜你，答对了。"
echo "中国的英文单词是${answer}。"
```

(2) 修改脚本的访问权限，执行脚本命令和结果如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ ./whileShell2.sh //输入脚本命令
中国的英文怎么写：
china //输入 China，然后按回车键
答错，请重新输入：
China //重新输入，输入 China，然后按回车键
恭喜你，答对了。
中国的英文单词是 China。
```

补充说明

null 在 shell 是一个空命令。使用 null 命令表达式，while 就成了一个死循环。若没有在循环体中，设置中断的语句，程序会一直执行。

3. until 循环语句

若理解了 while 的使用，那么就不难理解 until 语句。until 循环语句的含义是不断执行循环体的命令，直到判断表达式为真时退出循环。until 循环语句的语法格式如下所示：

```
until 判断表达式
do
循环体
done
```

当判断表达式执行的返回状态不为 0 执行循环体里的命令；若判断表达式的返回状态为 0 退出 until 循环。

例 21-12：使用 until 语句计算从 0 到指定数值的总和

(1) 编写 untilShell.sh 脚本，代码如下所示：

```
#!/bin/bash
echo "计算从 0 到指定数值的数值总和。"
echo "输入一个数："
read num      //从终端上获取一个用户输入的数值
resultNum=0 //定义总和结果变量
i=0          //定义循环变量
until [ $i -gt $num ] //表示直到 i 的值大于 num 时结束执行循环体
do    //until 循环体的开始
    resultNum=$(( ${resultNum} + $i )) //将当前循环变量的值加到总和变量上
    i=$(( $i + 1 ))      //循环变量加 1
done  //until 循环体的结束
echo "从 0 加到$num 的总和为$resultNum"
```

(2) 修改脚本的访问权限，执行脚本命令和结果显示如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ ./untilShell.sh
计算从 0 到指定数值的数值总和。
```

输入一个数：

9 //输入指定的数 9，然后按回车键

从 0 加到 9 的总和为 45

分析 until 循环语句的执行过程。脚本开始执行时，要求用户输入一个数值，若用户输入一个数值 9。在没有进入 until 之前，num 变量的值为 9，resultNum 和 i 的值都为 0。进入 until 循环语句，将 i 变量的值与 num 变量的值进行大于的比较，很明显此时的 i 值为 0，0 不大于 9，开始执行 until 循环体内的命令；经过循环体后 i 的值加 1，再次与 num 的值做比较，以此类推，直到 i 的值大于 num 的值，退出 until 循环。

21.2.4 循环控制

标准的循环语句都是执行完循环后才会结束循环。可编程时往往需要从循环中跳出来，或返回循环顶部，或者需要有某种办法中断死循环。Bash shell 提供了一组循环控制命令来处理这些情况，循环控制命令包括：shift 命令、break 命令和 continue 命令。

1. shift 命令

shift 命令将参数列表左移指定次数。在没有给定参数时，shift 命令把参数列表左移一次。一旦参数列表被移动，左端那个参数就从参数列表中删除了。while 循环遍历位置参数列表时，常用到 shift 命令。shift 命令的语法格式如下：

```
shift [n]
```

其中 n 是一个数值。

例 21-13：使用 shift 删除参数控制 while 循环

使用 shift 语句左移参数列表，同时在 while 循环中输出参数列表，shiftwhile.sh 脚本的编写和执行如下所示：

(1) 编写 shiftWhile.sh 脚本，代码如下所示：

```
#!/bin/bash
while [ $# -gt 0 ] // $# 表示用户输入的参数个数
```



```
do
    echo $*      // $*参数列表
    shift        //参数列表左移一次
done
```

(2) 修改脚本的访问权限, 执行脚本和结果显示如下所示:

```
dranyu@dranyu-desktop:~/Documents/program$ ./shiftWhile.sh a b c d e f g
a b c d e f g
b c d e f g
c d e f g
d e f g
e f g
f g
g
```

2. break 命令

break 命令用于强制退出循环, 但不退出程序。执行 break 命令后, 退出循环, 程序继续执行循环关键字 done 后的命令。在嵌套的循环中, break 命令使用执行跳出最内层的循环。在退出无限循环时很有用。break 命令的语法格式如下:

```
break [n]
```

其中 n 是数值, 表示跳出循环的层数。

例 21-14: 使用 break 命令跳出循环

将 while 循环嵌入到 for 循环中, 查找 for 循环中数值为 5 的趟次, 并显示在终端上, 找到后跳出 for 循环, 否则继续查找, breakShell.sh 的编写与执行如下:

(1) 编写 breakShell.sh 脚本, 代码如下所示:

```
1#!/bin/bash
2for i in 1 2 3 4 5 6 7 8 9      //for 循环
3do
4va=0
5while [ ${va} -lt ${i} ]          //for 循环里的 while 循环
6do
7if [ ${va} -eq 5 ]
8then
9echo "for 的循环${i}趟找到${va}。"
10break 2                         //使用 break 命令, 跳出 2 层循环, 即 while 循环和 for 循环
11fi
12va=$(( ${va} + 1 ))
13done
14echo "for 的循环${i}趟"          //在终端上显示信息, 其中${i}是 for 循环变量
15done
16echo "已跳出 for 循环。"
```

(2) 修改脚本的访问权限, 执行脚本和结果显示如下所示:

```
dranyu@dranyu-desktop:~/Documents/program$ ./breakShell.sh
for 的循环 1 趟
for 的循环 2 趟
for 的循环 3 趟
for 的循环 4 趟
for 的循环 5 趟
for 的循环 6 趟找到 5。
已跳出 for 循环。
```

为解说在源文件中添加了行号。从结果中, 我们可发现程序在跳出的时候没有执行第 14

行的命令，而是直接跳出了 for 循环，执行第 16 行的命令。也就是说，break 2 命令直接跳出了 2 层循环。

3. continue 命令

与 break 命令不同，continue 命令只是停止此趟执行过程，返回到循环语句的下一趟执行过程。continue 命令的语法格式如下：

```
continue [n]
```

其中 n 是数值，表示跳到的循环层次。

例 21-15：使用 continue 命令修改例 21-14 的脚本

在例 21-14 中，在 for 的每趟循环中查找数值 5，找到后，退出 while 循环，但过程跳到 for 循环的下一趟，continueShell.sh 脚本的编写与执行命令如下：

(1) 将 continueShell.sh 脚本的第 10 行命令修改为如下代码：

```
continue 2 //中止此趟 while 循环和 for 循环执行过程，返回到 for 的下一趟执行过程
```

(2) 修改脚本的访问权限，执行脚本命令和结果显示如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ ./continueShell.sh
for 的循环 1 趟
for 的循环 2 趟
for 的循环 3 趟
for 的循环 4 趟
for 的循环 5 趟
for 的循环 6 趟找到 5
for 的循环 7 趟找到 5
for 的循环 8 趟找到 5
for 的循环 9 趟找到 5
已跳出 for 循环。
```

从结果中可知，执行 continue 2 命令后，程序没有再往下执行 for 循环体中 continue 命令以下的其他命令，而是直接跳到了 for 循环的下一趟。

21.2.5 case 测试语句

case 测试语句是一个多分支的条件语句。如果一个程序有很多条件，若使用 if/elif/else 语句来完成很费劲。使用 case 语句可代替 if/elif/else 语句，实现多分支程序。case 测试语句的语法格式如下：

```
case 变量 in
    值 1)
        命令组 1
        ;;
    值 2)
        命令组 2
        ;;
    .....
    *)
        命令组 n
        ;;
esac
```

case 变量的值与值 1、值 2 等逐一比较，直至找到相同的值，然后执行后面定义的命令组代码，遇到双分号 “;” 结束此相应值定义的命令组，执行过程跳到 esac，退出 case 测试语句。



例 21-16：使用 case 测试语句编写一个选择菜单脚本

(1) 编写 caseShell.sh 脚本，代码如下所示：

```
#!/bin/bash
echo "你想要点水果吗？"
cat <<- FRUIT //cat 命令的 here 文档
    1) 苹果
    2) 香蕉
    3) 雪梨
    4) 橘子
FRUIT
echo "请选择你需要的水果类型(若需要苹果, 请输入 1): "
read choice //获取用户输入的水果类型代号
case ${choice} in
    1)
        echo "你要的是苹果."
        ;;
    2)
        echo "你要的是香蕉."
        ;;
    3)
        echo "你要的是雪梨."
        ;;
    4)
        echo "你要的是橘子."
        ;;
    *)
        echo "不好意思, 本店没有你需要的水果."
        ;;
esac
echo "欢迎下次光临!" //退出 case 语句后执行显示信息的命令
```

(2) 修改脚本的访问权限，执行脚本命令和结果显示如下所示：

```
dranyu@dranyu-desktop:~/Documents/programs$ chmod 755 caseShell.sh
dranyu@dranyu-desktop:~/Documents/programs$ ./caseShell.sh
你想要点水果吗？
1) 苹果
2) 香蕉
3) 雪梨
4) 橘子
请选择你需要的水果类型:
1 //输入 1 后, 按下回车键
你要的是苹果.
欢迎下次光临!
```

21.2.6 select 循环语句

here 文档是生成菜单的简便方法，而 Bash 提供了另一种循环机制实现菜单选择的功能，称为 select 循环。select 循环语句的语法格式如下：

```
select 变量 in 菜单列表
do
    命令组
done
```

select 语句会将菜单列表中的数据按照数字顺序排列成菜单项在标准错误输出上显示，

并显示 PS3 提示符请求用户；显示 PS3 提示符后，shell 等待用户输入，输入的是菜单项中的数字。PS3 提示符的默认值为 #?，同时也可在程序中进行设置。

例 21-17：使用 select 循环语句修改例 21-13 的脚本实现菜单选择的功能

(1) 编写 selectShell.sh 脚本，代码如下所示：

```
#!/bin/bash
PS3="请选择本店所提供的水果品种：" //设置 PS3 提示符
echo "你想要点水果吗？"
select choice in '苹果' '香蕉' '雪梨' '橘子' 'exit'
do
case ${choice} in
    苹果)
        echo "你要的是苹果。"
        ;;
    香蕉)
        echo "你要的是香蕉。"
        ;;
    雪梨)
        echo "你要的是雪梨。"
        ;;
    橘子)
        echo "你要的是雪梨。"
        ;;
    Exit)
        echo "欢迎下次光临！"
        break;; //break 语句是退出 select 循环的语句
    *)
        echo "不好意思，本店没有你需要的水果"
        ;;
esac
done
```

(2) 修改脚本的访问权限，执行脚本命令和结果显示如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ ./selectShell.sh
你想要点水果吗？
1) 苹果
2) 香蕉
3) 雪梨
4) 橘子
5) exit
请选择本店所提供的水果品种：1
你要的是苹果。
请选择本店所提供的水果品种：2
你要的是香蕉。
请选择本店所提供的水果品种：3
你要的是雪梨。
请选择本店所提供的水果品种：4
你要的是雪梨。
请选择本店所提供的水果品种：5
欢迎下次光临！
```

21.2.7 shell 内置命令

shell 有很多内置在其源代码中的命令。这些命令是内置的，所以 shell 不必到磁盘上搜



索它们，执行速度因此加快。Bash 提供的 help 功能，能提供任何内置命令的在线帮助，表 21-5 列出了这些内置命令。

表 21-5 Bash 内置命令

命 令	功 能
.	执行当前进程环境中的程序
. file	dot 命令从文件 file 中读取命令并执行
:	空操作，返回退出状态 0
alias	显示和创建已有命令的别名
bg	把作业放到后台
bind	显示当前关键字与函数的绑定情况，或将关键字与 readline 函数或宏进行绑定
break	从最内层循环跳出
break [n]	请参见 21.2.4 节“break 命令”
builtin [sh-builtin [args]]	运行一个内置 shell 命令，并传送参数，返回退出状态 0。如果一个函数与一个内置命令同名时，该命令将很有用
cd [arg]	改变目录，如果不带参数，则回到主目录，带参数则切换到参数所指的目录
command command [arg]	即使有同名函数，仍然执行该命令。也就是说，跳过函数查找
continue [n]	请参见 21.2.4 节“continue 命令”
declare [var]	显示所有变量，或用可选属性声明变量
dirs	显示当前记录的目录(pushd 的结果)
disown	从作业表中删除一个活动作业
echo [args]	显示 args 并换行
enable	启用或禁用 shell 内置的命令
eval [args]	把 args 读入 shell，并执行 args 指定的命令
exec command	运行命令，替换掉当前 shell
exit [n]	以状态 n 退出 shell
export [var]	使变量可被其他 shell 使用
fc	历史的修改命令，用于编辑历史命令
fg	把后台作业放到前台
getopts	解析并处理命令行选项
hash	控制用于加速命令查找的内部哈希表
help [command]	显示关于内置命令的有用信息。如果指定了一个命令，则将显示该命令的详细信息 21
history	显示带行号的命令历史列表
jobs	显示放到后台运行的作业
kill [-signal process]	向由 PID 号或作业号指定的进程发送信号。输入 kill -l 查看信号列表
let	用来计算算术表达式的值，并把算术运算的结果赋给变量
local	用在函数中，把变量的作用域限制在函数内部

续表

命 令	功 能
logout	退出当前的登录 shell
popd	从目录栈中删除项
pushd	向目录栈中增加项
pwd	打印出当前的工作目录
read [var]	从标准输入读取一行，保存到变量 var 中
readonly [var]	将变量 var 设为只读，不允许重置此变量
return [n]	从函数中退出，n 是指定 return 命令退出的状态值
set	设置选项和位置参量
shift [n]	将位置参量左移 n 次，请参见 21.2.4 节“shift 命令”
stop pid	暂停第 pid 号进程的运行
suspend	终止当前 shell 的运行，对登录 shell 无效
test	检查文件类型，并计算条件表达式
times	显示由当前 shell 启动的进程运行所累计用户时间和系统时间
trap [arg] [n]	当 shell 收到信号 n (n 为 0、1、2 或 15) 时，执行 arg 命令
type [command]	显示命令 command 的类型
typeset	同 declare。设置变量并赋予其属性
ulimit	显示或设置进程可用资源的最大限额
umask [八进制数字]	用户文件关于属主、属组和其他用户的创建模式掩码
unalias	取消所有的命令别名设置
unset [name]	取消指定变量的值或函数的定义
wait [pid#n]	等待 pid 号为 n 的后台进程结束，并报告它的结束状态

21.3 shell 函数



与其他编程语言一样，Bash 同样使用函数进行编程，尽管在某些实现方面上稍有限制。一个函数就是一个子脚本程序，是用于实现一系列操作的代码块。

21.3.1 函数简介

当存在重复代码的时候，或者当一个任务只需要轻微修改就被重复使用的时候，就需要考虑使用函数。函数的执行过程并不会创建新的 shell 进程，与当前调用该函数的 shell 脚本程序有相同的环境。

在使用函数之前，需要对函数进行定义；若函数不再被使用，可使清除函数；函数的使用一般限于当前 shell 环境，若使得当前 shell 下的子 shell 也可使用已定义的函数，则需要将已定义的函数导出。



- 定义函数

Bash 函数的语法格式如下：

```
function functionName
{
    statements
}
或
functionname ()
{
    statements
}
```

其中 `function` 是函数声明的关键字，`functionName` 是函数的名称。函数执行的最后结果正常返回是零；若发生错误，如函数已定义，或者语法错误发生，则会造成非零的返回值。函数的退出状态是由函数体内最后一次执行命令的返回状态所决定的。

- 清除函数

有时不需要一个已定义的函数，需要将函数删除。在 Bash 中提供了 `unset` 命令清除函数。格式为：

```
unset -f functionName
```

- 导出函数

Bash shell 还提供了函数的导出功能，其可将一个已定义的函数导出使子 shell 可以使用这些函数。其使用的格式为：

```
export -f functionName
```

其中 `export` 是导出关键字；`functionName` 是已定义函数的名称。

21.3.2 函数的参数和返回值

函数是在当前 shell 执行的，变量对函数和 shell 都是可见的。程序员在函数中对环境所做的任何改动也会改变 shell 的环境。

- 参数

我们知道在 shell 脚本中使用位置参数\$N 来传递值。在 shell 脚本定义的函数也是通过这种方式来传递值的。而且在函数里，位置参数是函数私有的，也就是说，函数对参数的操作不会影响在函数外使用的任何位置参数。

例 21-18：函数里的参数

在 `functionShell.sh` 脚本中编写一个 `hello` 函数，并调用 `hello` 显示相关的信息，脚本的编写内容和执行如下：

(1) 编写 `functionShell.sh` 脚本，代码如下所示：

```
1#!/bin/bash
2function hello( ) {          //定义 hello 函数
3echo "在函数里说: Hello,$*" //在 hello 函数上使用$*内部参数
4}
5hello Tom Jim               //调用 hello 函数, Tom 和 Jim 是 hello 函数的两个位置参数
6echo "在脚本里说: Hello,$*" //使用$*内部参数
```

(2) 修改脚本的访问权限，执行脚本和结果显示如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ ./functionShell.sh dranyu sunny
在函数里说: Hello, Tom Jim
在脚本里说: Hello,dranyu sunny
```

为方便解说，在脚本上添加了行号。执行 functionShell.sh 脚本命令时，将“dranyu”和“sunny”两个位置参数输入到脚本中，这样在脚本里\$*代表的值是“dranyu sunny”。可第3行代码中使用了\$*，却输入了“Hello,Tom Jim”的结果。由此可知，hello 函数使用的\$*变量的值是由调用 hello 函数时所传入的“Tom Jim”。也就是说，函数的位置参数是函数私有的，它的值是由调用函数时输入的参数决定的。

● 内置功能 local

使用 local 语句来定义一个函数私有的局部变量，此变量在函数退出后就自动消失，其使用的基本格式如下：

```
local 变量名
```

例 21-19：使用 local 定义一个局部变量

使用 local 语句在 hello 函数中定义一个局部变量 var，同时在 functionShell1.sh 脚本中定义了 a 和 var 变量。调用 hello 函数修改变量 a 和 var 变量的值，脚本的编写内容和执行结果如下：

(1) 编写 functionShell1.sh 脚本，代码如下所示：

```
1#!/bin/bash
2a=0 //定义变量 a，并赋值为 0
3var=2 //定义变量 var，并赋值为 2
4function hello() { //定义 hello 函数
5local var //local 语句将 var 变量声明为局部变量
6var=$1 //为 var 变量赋值
7a=1 //为 a 变量赋值
8echo "在函数里说: Hello, ${var}。编号: ${a}"
9}
10hello $* //设置 hello 的参数是$*变量的值，即运行此脚本时输入的参数列表
11echo "在脚本里说: Hello, ${var}。编号: ${a}"
```

(2) 修改脚本的访问权限，执行脚本和结果显示如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ ./functionShell1.sh sunny
在函数里说: Hello,sunny。编号: 1
在脚本里说: Hello,2。编号: 1
```

由执行显示结果可知，使用 local 在 hello 函数声明的 var 局部变量在 hello 函数中重新赋值为 \$1 变量的值，可是脚本里的 var 变量值并未被修改；而没有使用 local 定义的 a 变量在 hello 函数被修改后，其在脚本中的值也被修改了。

● 内置功能 return

return 命令用于退出函数并将执行过程的控制转回程序调用函数的位置，也就是说跳出函数，而不退出脚本程序。如果没有特别指定参数，函数的返回值其实就是函数中最后一条命令的退出状态。如果给 return 命令赋一个值，该值就被保存在状态变量 \$? 中。

例 21-20：使用 return 返回函数的值

在 square 函数中计算位置参数 \$1 的平方值，并通过 return 语句将计算的数值保存到当前 shell 脚本的状态变量 \$? 中，functionShell2.sh 脚本的编写内容和执行结果如下：



(1) 编写 functionShell2.sh 脚本，代码如下所示：

```
#!/bin/bash
square(){ //定义函数 square
    return $(( $1 * $1 )) //使用 return 返回$(( $1 * $1 ))的算术值,
    $1 表示调用 square 函数时输入的参数
}
echo "求指定数的平方，输入一个数："
read num //从终端上获取一个用户输入的值，将其保存到 num 变量中
square ${num} //调用 square 函数，参数为由 read 命令获取的 num 变量的值
echo "${num}的平方为: $?" //显示前一个命令的保存在状态变量中的值,
前一个命令为调用 square 函数
```

(2) 修改脚本的访问权限，执行脚本和结果显示如下所示：

```
dranyu@dranyu-desktop:~/Documents/program$ ./functionShell2.sh
求指定数的平方，输入一个数：
12
12 的平方为: 144
```

从结果可知，return 命令将其后的值保存在状态变量\$?中。在调用 square 函数后，就应该将保存在状态变量\$?里的值提取出来，以免执行其他命令将其替换。

补充说明

`$(($1 * $1))`表达式表示进行\$1 变量值与\$1 变量值的乘法运算。`$((算术表达式))`语法可执行相应的算术计算。

21.4 shell 脚本综合实例



shell 中提供了多种命令可查看系统的各类信息。若希望使用一个脚本就可显示系统详细的信息，可使用如例 21-18 中的脚本代码。只要将代码复制到编辑器中，然后生成一个可执行文件，即可运行此脚本。

例 21-21：显示系统的信息

在使用和维护计算机系统时，需要清楚地了解当前系统的信息，脚本 sysInfo.sh 的内容如下：

```
1#!/bin/bash
2#
3# Linux Shell Scripting Tutorial 1.05r3, Summer-2002
4#
5# Written by Vivek G. Gite <vivek@nixcraft.com>
6#
7# Latest version can be found at http://www.nixcraft.com/
8#
9# Changed by dranyu <dranyu@gmail.com> 2010.1.5
10#
11
12 nouser=`who | wc -l` //将 who | wc -l 命令的字符串赋值给 nouser 变量
13 echo -e "用户名: $USER (Login name: $LOGNAME)" >> /tmp/info.tmp.01.$$$
14 echo -e "当前使用的 Shell: $SHELL" >> /tmp/info.tmp.01.$$$
15 echo -e "当前用户的主文件夹: $HOME" >> /tmp/info.tmp.01.$$$
```

```

16 echo -e "当前操作系统类型: $OSTYPE" >> /tmp/info.tmp.01.$$$
17 echo -e "PATH: $PATH" >> /tmp/info.tmp.01.$$$
18 echo -e "当前工作目录: `pwd`" >> /tmp/info.tmp.01.$$$
19 echo -e "当前登录系统的人数: $nouser user(s)" >> /tmp/info.tmp.01.$$$
20
21 if [ -f /etc/redhat-release ]
22 then
23   echo -e "操作系统: `cat /etc/redhat-release`" >> /tmp/info.tmp.01.$$$
24 fi
25
26 if [ -f /etc/shells ]
27 then
28   echo -e "支持的 Shell 类型: " >> /tmp/info.tmp.01.$$$
29   echo -e "`cat /etc/shells`" >> /tmp/info.tmp.01.$$$
30 fi
31
32 if [ -f /etc/sysconfig/mouse ]
33 then
34
35   echo -e "-----" >> /tmp/info.tmp.01.$$$
36   echo -e "计算机鼠标信息: " >> /tmp/info.tmp.01.$$$
37   echo -e "`cat /etc/sysconfig/mouse`" >> /tmp/info.tmp.01.$$$
38 fi
39 echo -e "-----" >> /tmp/info.tmp.01.$$$
40 echo -e "计算机CPU 信息:" >> /tmp/info.tmp.01.$$$
41 echo -e "-----" >> /tmp/info.tmp.01.$$$
42 cat /proc/cpuinfo >> /tmp/info.tmp.01.$$$
43
44 echo -e "-----" >> /tmp/info.tmp.01.$$$
45 echo -e "计算机内存信息:" >> /tmp/info.tmp.01.$$$
46 echo -e "-----" >> /tmp/info.tmp.01.$$$
47 cat /proc/meminfo >> /tmp/info.tmp.01.$$$
48
49 if [ -d /proc/ide/hda ]
50 then
51
52   echo -e "硬盘信息:" >> /tmp/info.tmp.01.$$$
53   echo -e "-----" >> /tmp/info.tmp.01.$$$
54   echo -e "模块: `cat /proc/ide/hda/model`" >> /tmp/info.tmp.01.$$$
55   echo -e "驱动: `cat /proc/ide/hda/driver`" >> /tmp/info.tmp.01.$$$
56   echo -e "缓存大小: `cat /proc/ide/hda/cache`" >> /tmp/info.tmp.01.$$$
57 fi
58 echo -e "-----" >> /tmp/info.tmp.01.$$$

```



```
/tmp/in  fo.tmp.01.$$$
59 echo -e "已加载的文件系统信息:" >> /tmp/info.tmp.01.$$$
60 echo -e "-----" >>
/tmp/in  fo.tmp.01.$$$
61 cat /proc/mounts >> /tmp/info.tmp.01.$$$
62
63 if which gdialog > /dev/null
64 then
65     gdialog --backtitle "Linux Software Diagnostics (LSD) Shell Script Ver.3.2"
--title  "按下 Up/Down 键进行滚动" --textbox /tmp/info.tmp.01.$$$ 70 70
66 else
67     cat /tmp/info.tmp.01.$$$ |more
68 fi
69
70 rm -f /tmp/info.tmp.01.$$$
71
72 #
73 # ./ch.sh: vivek-tech.com to nixcraft.com reference converted using this tool
74 # See the tool at http://www.nixcraft.com/unixlinuxfeatures/tools/
75 #
```

对 sysInfo.sh 脚本的解释如下：

- (1) 第 1 行是 Bash shell 的开始，指定使用的 shell 路径。
- (2) 第 2 行到第 10 行是文件的版本说明。
- (3) 第 12 行说明在脚本上。
- (4) 第 13 行到第 19 行，将系统基本的信息添加到/tmp/info.tmp.01.\$\$临时文件上。在 \$\$中，前两个构成了系统的内部变量显示的是脚本运行时的进程 ID，而最后那个\$只是作为 \$字符标识。>>是 shell 中的一个追加文件内容的符号，表示将 echo 显示的内容重定向到 /tmp/info.tmp.01.\$\$临时文件。
- (5) 第 21 行到第 24 行，测试 redhat 的版本。
- (6) 第 26 行到第 30 行，将系统所支持的 shell 类型信息追加到/tmp/info.tmp.01.\$\$临时文件。
- (7) 第 32 行到第 38 行，将计算机鼠标信息追加到/tmp/info.tmp.01.\$\$临时文件。
- (8) 第 39 行到第 42 行，将计算机 CPU 信息追加到/tmp/info.tmp.01.\$\$临时文件。
- (9) 第 44 行到第 47 行，将计算机内存信息追加到/tmp/info.tmp.01.\$\$临时文件。
- (10) 第 49 行到第 57 行，将计算机硬盘信息追加到/tmp/info.tmp.01.\$\$临时文件。
- (11) 第 58 行到第 61 行，将计算机已加载的文件系统信息追加到/tmp/info.tmp.01.\$\$临时文件。到此为止，我们所需要的系统信息都添加到了/tmp/info.tmp.01.\$\$临时文件里。
- (12) 第 63 行到第 68 行的代码表示选择显示系统信息的方式。若系统有 gdialog 程序，那么使用窗口的形式显示/tmp/info.tmp.01.\$\$临时文件里的内容；否则使用 cat 命令在终端上显示此临时文件的内容。

sysInfo.sh 脚本的执行结果如图 21-1 所示。

图 21-1 sysInfo.sh 的执行结果

21.5 小结



Bash shell 的使用可更方便地管理和维护系统。Ubuntu 系统的服务多以 Bash 脚本的方式启动。本章详细讲述了 Bash shell 脚本的编写语法，讲述了 shell 中函数的应用。通过本章的学习，Ubuntu 系统管理员能够掌握阅读 Bash shell 脚本的能力，掌握编写 shell 脚本管理和维护系统的能力。

21.6 习题



1. 简述编写和运行一个 shell 脚本的过程。
2. 编写一个 shell 脚本命名为 showInfo.sh，将 shell 中系统变量的值显示到终端（提示：系统变量在 21.2.1 节已讲述）。
3. 编写 userShell1.sh 脚本，定义一个 var 变量，并分别使用双引号和单双号将 var 变量的引用括起来显示在终端上。
4. 编写 envShell.sh 脚本显示当前 shell 环境变量的值。
5. 编写 checkShell.sh 脚本判断文件 file1 是否存在，若存在则将 file1 的内容复制到文件 file2 中；若不存在则创建 file1 文件，并向文件 file1 中插入“file1 文件被创建。”内容。
6. 使用循环语句编写 haha.sh 脚本，实现如下图 1 的模式输出。

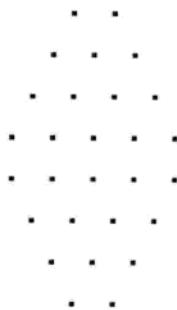
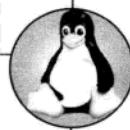


图 1 习题 6

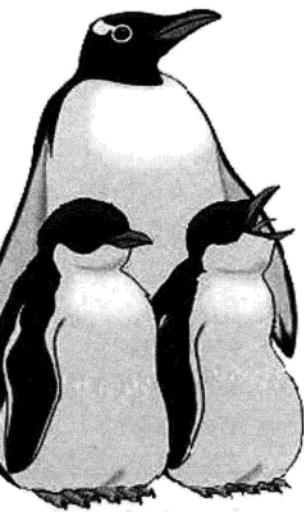
7. 使用 `select` 命令编写一个带有菜单的 `menu.sh` 脚本，菜单项为：语文、数学、英语、计算机和思想政治。
8. 编写一个计算 10 的阶乘脚本 `cal.sh`。
9. 使用函数的知识，编写计算一个输入长宽的数值求面积的脚本 `quadrature.sh`。
10. 编写一个 `addSysUser.sh` 的脚本，实现添加系统用户的功能。思路：从终端获取用户的名称、密码、电话、住址和真名等，将密码进行加密，将用户的相关信息追加到 `passwd` 文件中。
11. 编写一个 `clock.sh` 脚本实现数字时钟。思路：写一个 `while` 死循环，然后每执行一次 `while` 循环就使程序休眠 1 秒钟，更新时间再显示。

Linux

第22章

版本控制工具

现在的软件项目开发中，必然涉及版本控制（Revision Control）工具。没有使用版本控制工具的开发过程，有人形容就如同生活在“黑暗时代”。版本控制工具是项目开发中必不可少的，以此进行的版本控制可以确保在软件项目开发中，不同的开发人员所涉及的同一文档或代码都得到修改和更新。本章主要进述如何构建 Subversion 版本控制服务器，以及如何使用 Subversion 客户端管理程序代码。





22.1 版本控制简介

版本控制的目的是实现开发团队并行开发、提高开发效率的基础。其目的在于对软件开发进程中文件或目录的发展过程提供有效的追踪手段，保证在需要时可回到旧的版本，避免文件的丢失、修改的丢失和相互覆盖，通过对版本库的访问控制避免未经授权的访问和修改，达到有效保护企业软件资产和知识产权的目的。目前，广泛流行的版本控制软件包括 SVN 和 CVS 版本控制工具。

22.1.1 CVS 简介

CVS 是并发版本系统（Concurrent Versions System）的英文缩写，是主流的开放源码网络透明的版本控制系统。从 CVS 的起因和特点介绍 CVS 版本控制系统。

开源软件的一个重要特点就是代码的获得和改进无地域限制，这是传统软件开发所不具备的。这意味着开源软件的开发是在全球进行的项目，每个人都可加入到项目中。一个地域上分散的志愿者组织显然不能投入很多的时间来训练其成员彼此合作，这就需要一套项目管理办法，确保新成员能较容易地适应工作。同时有一个自动的机制接收外来代码，使每个成员能及时得到最新修改的代码。当然，这不仅仅是开源社区的需要，只是开源社区的人员分散、资源不易管理，更需要一个版本管理工具。这样的一个工具首先应该支持世界范围的协作，保持发布版本的一致性。其次，它能够容易地汇集各个特定版本的 Bug，并在全球范围同步一致地进行修改；再者它需要使任何一个开发者能够追踪软件的变化；而在开发者为软件增加新功能的同时，不能妨碍一般用户使用一个相对稳定的版本。

CVS 很好地解决了这一问题，除了让开发人员可以使用 CVS 很容易地把代码变化存入代码库之外，它还为不同角色的人员设置了不同的访问权限。例如，不需要修改代码的人员可以以匿名只读方式访问代码仓库；而需要修改代码的开发人员每个人都能在自己的机器上建立一个开发树或工作副本，当需要在一个特殊的代码区工作时，首先通过简单的命令，更新本地的开发树或工作副本，以保持全球范围开发状态的一致。这样，就可以避免出现这样的问题：花了半天时间修改的 Bug，在提交的时候发现别人已经解决了。一旦确认代码中的 Bug 别人还没解决，开发者可以马上开始工作。当这些问题解决后，CVS 会自动产生补丁，并将补丁发送给维护人员进行检查，最后可能将补丁并入到主项目树中去。

尽管 CVS 的功能和使用方法看起来颇为庞大复杂，其实 CVS 最重要的只有两点，即记录保存和协作。人们有时要将一个程序的当前状态与先前某一状态做比较。例如，在为程序添加新功能的过程中，有人可能会通报试用版的 Bug。为了找到问题之所在，一方面程序必须可以找回原来指定时间的可用状态。事实上，开发者可以简单地说一句，把以前程序的状态给我，或者说把最新公开发行的版本给我，这就是 CVS 对历史记录的保存作用。另一方面，CVS 系统要顺利工作，开发者必须彼此知道在某个时刻准备做什么工作。CVS 能够在代码提交的时候提醒开发者代码是否存在冲突；当某人因为权限问题不能工作的时候，彼此能很快地沟通。这就是 CVS 协同工作、跟踪冲突的能力。CVS 的一般任务主要是访问已有的代码库、

创建新项目、检出工作复制、进行更改、检测并解决冲突、浏览记录信息、检查并还原更改等。

CVS 现已成为广泛使用的版本管理系统，普遍应用在软件开发过程中，是基本的软件工程配置管理工具之一。不论是一个庞大的工程，还是个人开发一个小软件，都可以方便地使用 CVS 来管理开发过程，提高效率，方便管理。

22.1.2 SVN 简介

Subversion，简称 SVN，是一个开放源代码的版本控制系统，相对于 CVS，其采用了分支管理系统，它的被设计目标就是取代 CVS。互联网上越来越多的控制服务从 CVS 转移到 Subversion。

2000 年初，开发人员要写一个 CVS 的自由软件代替品，它保留 CVS 的基本思想，但没有它的错误和局限。当时许多开发人员也发现了 CVS 的不足，Subversion 项目发起后，很快就获得了这部分人的支持和参与。

虽然，在 2006 年 Subversion 的使用族群仍然远少于传统的 CVS，但已经有许多开放源码团体决定将 CVS 转换为 Subversion。已经转换使用 Subversion 的团体包括了 FreeBSD、Apache Software Foundation、KDE、GNOME、GCC、Python、Samba 和 Mono 等。

SVN 继承了 CVS 的部分功能：代码版本的存储与维护；代码版本的追踪回溯；代码版本的分合控制；开发者多人合作开发专案；远端管理维护代码；代码匿名获取。

SVN 优于 CVS 之处如下：

(1) 统一的版本号。CVS 是对每个文件顺序编排版本号，在某一时间各文件的版本号各不相同。而 Subversion 下，任何一次提交都会对所有文件增加到同一个新版本号，即使是提交并不涉及的文件。所以，各文件在某任意时间的版本号是相同的。版本号相同的文件构成软件的一个版本。

(2) 原子提交。一次提交不管是单个还是多个文件，都是作为一个整体提交的。在这当中发生的意外例如传输中断，不会引起数据库的不完整和数据损坏。

(3) 重命名、复制、删除文件等动作都可保存在版本历史记录当中。

(4) 对于二进制文件，使用了节省空间的保存方法。

(5) 记录目录的版本历史。整个目录树可以被移动或者复制，操作很简单，而且能够保留全部版本记录。

(6) 分支的开销非常小。

(7) 优化过的数据库访问，使得一些操作不必访问数据库就可以做到。这样减少了很多不必要的与数据库主机之间的网络流量。

(8) 支持元数据管理。每个目录或文件都可以定义属性，它是一些隐藏的键值对，用户可以自定义属性内容，而且属性和文件内容一样在版本控制范围内。

22.2 构建 svn 服务器

构建 svn 服务器



Subversion 版本控制提供了多种访问方式，包括 Web 服务访问方式、svn 服务访问方式



和本地服务访问方式。其中 svn 服务访问方式和本地服务访问方式在安装 Subversion 软件包后就自动添加了。若要使用 Web 服务访问方式，系统管理员还需要安装 Web 服务器和相应的连接模块。而且 svn 的 Web 服务访问方式是目前使用比较广泛的方式。

22.2.1 安装 Subversion

目前，Subversion 软件包已经包含在 Ubuntu 的 main 仓库中，系统管理员可从仓库中安装 Subversion 软件包。在新立得软件包管理器中，搜索 Subversion 软件包，然后选中并安装此软件包。同样，系统管理员也可在终端上输入如下命令安装 Subversion 软件包：

```
root@dranyu-desktop:~# apt-get install subversion
```

安装完毕后，可使用如下命令查看系统所安装的 Subversion 软件包版本号，以及其支持的访问模块：

```
root@dranyu-desktop:~# svn --version          //注释：此为 svn 的一个显示版本信息的命令
svn, 版本 1.5.4 (r33841)                      //显示 svn 的版本
      编译于 Aug 7 2009, 01:44:11
      版权所有 (C) 2000-2008 CollabNet。
      Subversion 是开放源代码软件, 请参阅 http://subversion.tigris.org/ 站点。
      此产品包含由 CollabNet(http://www.Collab.Net/) 开发的软件。
      可使用以下的版本库访问模块：
* ra_neon : 通过 WebDAV 协议使用 neon 访问版本库的模块。
  - 处理 “http” 方案
  - 处理 “https” 方案
* ra_svn : 使用 svn 网络协议访问版本库的模块。- 使用 Cyrus SASL 认证
  - 处理 “svn” 方案
* ra_local : 访问本地磁盘的版本库模块。
  - 处理 “file” 方案
```

此时 Subversion 软件包已被安装到系统中。

在使用 apache2 作为 Subversion 的 web 服务器时，必须确定 apache2 服务器已经安装到系统中，而且还需要添加一个使用 apache2 和 subversion 通信的模块，执行如下命令安装此通信模块软件包：

```
root@dranyu-desktop:~# apt-get install libapache2-svn
```

此通信模块软件包包括了 mod_dav_svn 和 mod_authz_svn 模块。mod_dav_svn 模块为 svn 提供 webDAV 协议支持，而 mod_authz_svn 模块为 svn 提供了用户验证功能。通过这些模块项目的开发人员就可通过 apache2 的 web 服务来访问 svn 的数据。

补充说明

apache2 服务器的构建在第 18 章已做详述。一般来说，在网页浏览器上，输入 `http://localhost` 就可测试 apache2 服务器是否安装。如果已安装，网页显示“`It works`”字符串。

22.2.2 创建 svn 数据仓库

许多位置都可以放置 Subversion 数据仓库，其中两个最常用的是：`/usr/local/svn` 以及 `/home/svn`。为了在下面的描述中简单明了，我们将 Subversion 数据仓库存放在 `/home/svn` 目录下，并且假设拥有一个名为“`web3ds`”的项目。

1. 准备工作

为网络用户和本地用户能够拥有对数据仓库文件的读写权限，一般来说，需要创建一个名为 Subversion 的组，并将系统的相关用户设置为 Subversion 组的成员，创建 svn 数据仓库的前期准备如下。

(1) 创建组和设置添加组成员的命令如下所示：

```
root@dranyu-desktop:~# addgroup subversion //添加的命令
正在添加组 'subversion' (GID 1006)...
完成。
root@dranyu-desktop:~# usermod -G subversion -a www-data //添加 www-data 组员
root@dranyu-desktop:~# usermod -G subversion -a dranyu //添加 dranyu 组员
root@dranyu-desktop:~# cat /etc/group | grep subversion //显示 subversion 组的组员
subversion:x:1006:www-data,dranyu //www-data 和 dranyu 是 subversion 组的组员
```

补充说明

www-data 是一个在安装 apache2 服务器时系统自动添加的用户。从网络上连接到 apache2 服务器的用户都会拥有与 www-data 用户相同的权限。

(2) 创建保存 svn 所有仓库的根目录/home/svn 和项目目录/home/svn/web3ds 并修改此目录的访问权限，使用 root 用户和 Subverison 组的组员拥有读写权限，命令执行和结果显示如下所示：

```
root@dranyu-desktop:/home# mkdir svn //注释：创建 svn 目录
root@dranyu-desktop:/home# cd svn //激忿进入 svn 目录
root@dranyu-desktop:/home/svn# mkdir web3ds //创建 myproject 目录
root@dranyu-desktop:/home/svn# ls -l //查个 web3ds 目录的访问权限
总用量 4
drwxr-xr-x 2 root root 4096 2009-12-11 13:52 web3ds

//修改 web3ds 目录的访问权限，设置 web3ds 所属的组为 subversion
root@dranyu-desktop:/home/svn# chown -R root:subversion web3ds
root@dranyu-desktop:/home/svn# ls -l //显示修改组后 web3ds 的访问权限变化
总用量 4
drwxr-xr-x 2 root subversion 4096 2009-12-11 13:52 web3ds
```

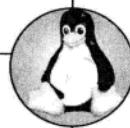
2. 创建数据仓库 web3ds

做好了准备工作后，还需要使用 Svnadmin 创建数据仓库，并为 Subversion 组设置相应的权限。svnadmin 创建 svn 数据仓库的命令格式如下：

svnadmin create 数据仓库目录

若命令的数据仓库目录不存在，系统会为用户自动添加一个。在本实例中，创建 /home/svn/web3ds 数据仓库的命令执行和结果显示如下所示：

```
root@dranyu-desktop:/home/svn# svnadmin create web3ds //注释：创建 web3ds 数据仓库
root@dranyu-desktop:/home/svn# cd web3ds //进入 web3ds 目录
root@dranyu-desktop:/home/svn/web3ds# ls -l //查看 web3ds 目录下的所有文件
总用量 24
drwxr-xr-x 2 root root 4096 2009-12-11 14:14 conf
drwxr-sr-x 6 root root 4096 2009-12-11 14:14 db
-r--r--r-- 1 root root 2 2009-12-11 14:14 format
drwxr-xr-x 2 root root 4096 2009-12-11 14:14 hooks
drwxr-xr-x 2 root root 4096 2009-12-11 14:14 locks
-rw-r--r-- 1 root root 229 2009-12-11 14:14 README.txt
```



其中 web3ds 里的所有文件是由 svnadmin 在创建数据仓库时所创建的。其中主要的两个文件目录是 conf 和 db 目录。conf 目录保存了 svn 用户的用户验证、用户密码和默认 svn 访问方式的服务器配置。db 目录是用来保存 web3ds 项目进行中文件信息与项目开发的版本控制信息。我们可发现 web3ds 目录下所有文件所属的用户是 root，其所属的组也是 root，那么其他 Ubuntu 的系统用户是无法编辑 conf 目录里的文件的。所以，需要执行如下命令实现 dranyu 用户和 www-data 用户拥有对 web3ds 仓库的配置文件进行修改的权限：

```
root@dranyu-desktop:/home/svn# chmod -R g+rws web3ds //g 表示修改的是组成员的权限配置  
root@dranyu-desktop:/home/svn# ls -l web3ds //显示修改组权限的文件信息  
总用量 24  
drwxrwsr-x 2 root root 4096 2009-12-11 14:39 conf // drwxrwsr-x 字段中的 rws 表示组员的  
拥有读写权限  
.....
```

可是，此时 web3ds 里边的文件所属的组是 root，可命令如下命令，将所有文件的所属组修改为 Subversion：

```
root@dranyu-desktop:/home/svn# chown -R root:subversion web3ds
```

3. 测试

这样完成 svn 数据仓库的创建。此时，可以使用默认支持的 svn 访问方式——file 本地访问方式访问 web3ds 数据仓库。使用 dranyu 的身份获取 web3ds 数据仓库里的代码文件，在 dranyu 的主文件夹中创建一个 project 的目录，进入 project 目录，然后执行 svn 的取出命令，将 svn 服务器上的 web3ds 项目的代码数据读取到 project，执行命令和结果显示如下所示：

```
dranyu@dranyu-desktop:~$ mkdir project //创建 project 目录  
dranyu@dranyu-desktop:~$ cd project //进入 project 目录  
dranyu@dranyu-desktop:~/project$ dir //显示文件列表  
dranyu@dranyu-desktop:~/project$ svn co file:///home/svn/web3ds //执行 svn 的检出命  
令，以  
file:/// 的访问方式访问  
取出版本 0。  
dranyu@dranyu-desktop:~/project$ ls -l //显示文件列表  
总用量 4  
drwxr-xr-x 3 dranyu dranyu 4096 2009-12-11 15:00 web3ds
```

从结果中，可发现执行 svn co 命令后，web3ds 项目的根文件目录已从 svn 服务器上保存到了本地/home/dranyu/project 目录下。这表示数据仓库创建成功，只是数据仓库里目前还没有数据代码。

4. 导入数据到数据仓库

一般地，在创建了空的数据仓库后，需要将项目的第一版本导入到数据仓库中。假设原始项目的内容在 /home/project 目录下。在 svn 系统中，数据仓库的根目录下包含 trunk、branches 和 tags 目录，其中 trunk 表示项目开发的主线，保存了项目的完整版本；branches 表示项目开发的支线，保存了在项目开发过程中开发人员从 trunk 目录提取的一个独立功能模块或保存开发者相关功能试验的项目；tags 目录保存一个比较完整的稳定的版本，是对 trunk 中的版本的一个标记版本。project 目录也就包含了这几个 svn 的主要文件目录。将 project 目录里的内容导入到 web3ds 数据仓库中，命令如下所示：

```
//注释：使用了 svn import 命令。“.”表示当前目录下的所有内容，file:///home/svn/web3ds 表示 web3ds  
数据仓库。“--username dranyu”表示以 dranyu 的身份执行操作  
root@dranyu-desktop:/home/project# svn import . file:///home/svn/web3ds --username
```

```
dranyu
认证领域: <http://localhost:80> web3ds subversion repository
“dranyu”的密码: //输入 dranyu 的 svn 密码
//以下显示导入到 web3ds 数据仓库的文件夹或文件
增加      trunk
增加      trunk/doc
增加      trunk/src
增加      branches
增加      tags

提交后的版本为 1。 //显示 web3ds 数据仓库的版本号
//使用 svn list 命令浏览 web3ds 数据仓库根目录的内容
root@dranyu-desktop:/home/project# svn list http://localhost/svn/web3ds --username
dranyu
branches/
tags/
trunk/
```

结果显示了，文件已导入到 web3ds 数据仓库。

22.2.3 配置 apache 的 svn 模块

安装了 libapache-svn 软件包后，相关的 apache2 模块配置文件 `dav_svn.conf` 就会添加到 `/etc/apache2/mods-available/` 目录下。若要使 svn 使用 apache2 的网络服务，还需要在配置文件 `dav_svn.conf` 中添加如下内容：

```
<Location /svn > //设置数据仓库的网络访问路径为 “http://localhost/svn/项目名”
DAV svn          //使用的模块为 svn 模块
SVNParentPath /home/svn //设置所有数据仓库在本地磁盘上的存储位置
AuthType Basic      //设置用户访问数据仓库时使用的验证方式为 Basic
AuthName "web3ds subversion repository" //设置验证界面的标题
AuthUserFile /etc/subversion/passwd //设置验证用户的密码文件 passwd
Require valid-user //设置对数据仓库所做的任何操作都需要验证
</Location> //结束标签
```

其中 `passwd` 密码文件是由 apache2 服务器中的 `htpasswd` 命令生成的文件。

1. 设置用户密码

在默认情况下，并没有 `/etc/subversion/passwd` 文件。在重启 apache2 服务器前，需要使用 `htpasswd` 命令添加 `passwd` 文件。`htpasswd` 命令创建密码文件的基本语法如下：

```
htpasswd [-c] 密码文件 用户名
```

例 22-1：创建密码文件

创建 `passwd` 文件并向 `passwd` 文件添加 `dranyu` 用户，命令如下所示：

```
root@dranyu-desktop:/etc/subversion# htpasswd -c passwd dranyu //注释: -c 表示创建一个
新的密码文件 passwd
New password: //输入用户 dranyu 的密码，然后按下回车键
Re-type new password: //再次输入用户 dranyu 的密码，然后按下回车键
Adding password for user dranyu
root@dranyu-desktop:/etc/subversion# ls -l      //显示文件列表
总用量 16
-rw-r--r-- 1 root root 5786 2009-08-07 10:26 config
-rw-r--r-- 1 root root 21 2009-12-11 18:39 passwd //新添的密码文件
-rw-r--r-- 1 root root 3619 2009-08-07 10:26 servers
root@dranyu-desktop:/etc/subversion# cat passwd //查看 passwd 密码文件的内容
```



Ubuntu Linux 从初学到精通

```
dranyu:r2NsxKXELJgos // “:” 将用户名和密码隔开，密码已加密
```

例 22-2：向 passwd 添加新用户

向 passwd 密码文件中新添一个 sunny 的用户，执行命令和结果显示如下所示：

```
root@dranyu-desktop:/etc/subversion# htpasswd passwd sunny //添加用户密码对的命令
New password: //输入 sunny 用户的密码，然后按下 Enter 回车键
Re-type new password: //再次输入 sunny 用户的密码，然后按下 Enter 回车键
Adding password for user sunny
root@dranyu-desktop:/etc/subversion# cat passwd //查看 passwd 密码文件中的内容
dranyu:r2NsxKXELJgos
sunny:.0siRSKpgCIUs //新添的 sunny 用户和密码
```

2. 重启 apache2 服务器

修改了 /etc/apache2/mods-available/dav_svn.conf 配置文件后，需要重启 apache2 服务器，这样才能使配置起作用。重启 apache2 服务器的执行命令和结果显示如下：

```
root@dranyu-desktop:/etc/apache2/mods-enabled# /etc/init.d/apache2 restart
//restart 选项表示
重启 apache2 服务器
* Restarting web server apache2                                         apache2: Could
not reliably determine the server's fully qualified domain name, using 127.0.0.1 for
ServerName
... waiting apache2: Could not reliably determine the server's fully qualified domain
name, using 127.0.0.1 for ServerName
[ OK ]
```

3. 测试 svn 的 http 访问方式

在上一小节中，我们使用了 file 本地访问的文件取出 svn 数据仓库的代码，现在使用 http 的方式取出 svn 数据仓库的代码，命令如下：

```
dranyu@dranyu-desktop:~/project$ svn co http://localhost/svn/web3ds --username dranyu
认证领域: <http://localhost:80> web3ds subversion repository
“dranyu”的密码: //输入 dranyu 用户 svn 的密码
取出版本 0。 //显示当前所取出项目版本号为 0
```

其中 http://localhost/svn/web3ds 表示请求访问在 /home/svn 目录下名为 web3ds 的项目数据仓库。“--username dranyu” 表示以 dranyu 用户的身份进行操作。在命令输入后，svn 要求输入 dranyu 的用户密码。

从上述结果可知，svn 已可使用 http 的服务，并拥有 Basic 基本类型的用户验证机制。

22.2.4 配置用户权限

尽管设置了用户的密码，可是 svn 服务器上项目并不止一个，往往服务器上拥有多个同时开发的项目。这时，服务器就必须为开发者设置项目组，并为项目组上的成员分配不同的访问权限。配置 svn 用户的权限必须设置用户权限管理文件，在上节的 dav_svn.conf 配置文件的 <Location /svn></Location> 标签之间添加如下代码：

```
AuthzSVNAccessFile /etc/subversion/authz //authz 为用户权限管理文件
```

使得 dav_svn.conf 配置文件包含如下内容：

```
<Location /svn>
DAV svn
SVNParentPath /home/svn
AuthType Basic
AuthName "web3ds subversion repository"
```

```

AuthUserFile /etc/subversion/passwd
AuthzSVNAccessFile /etc/subversion/authz
Require valid-user
</Location>

```

1. 用户权限文件的编写

authz 文件的编写可参考/home/svn/web3ds/conf/authz 文件的编写格式, 基本编写格式如下:

```

[节名]
选项 1=选项值 1
...
选项 n=选项值 n

```

节名的方式和含义有以下几种。

- [alias]节: 此节表示用户使用的别名, 也就是说用户使用不同的别名登录到 svn 数据仓库中, svn 会将用户识别为同一个人。
- [groups]节: 此节定义项目组, 并且定义项目组的成员。
- [路径]节: 此节定义以/home/svn 为根目录的数据仓库目录, 如[/]表示/home/svn 目录, 而[/web3ds]表示数据仓库 web3ds 的目录/home/svn/web3ds。
- [数据仓库: 路径]节: 数据仓库定义的是数据仓库的名称, 路径是以此数据仓库为根目录的路径。如[web3ds: /]表示 web3ds 数据仓库的根目录, 而[web3ds: /doc]表示 web3ds 数据仓库上的 doc 目录。

在不同的节中, 选项的含义不同, 在各节中的含义如下。

- [alias]节: 选项表示的是一个简写用户名, 而选项值是用户名的别名。如, jeo=Joe Average, 用户名 jeo 的别名为 Joe Average。这两个名字同表示一个人。
- [groups]节: 选项表示的是一个组名, 而选项值是组的成员名。如“pro1=dranyu,sunny”表示定义一个包含了 dranyu 和 sunny 两个开发者的项目组 pro1。
- [路径]节和[数据仓库: 路径]节: 选项表示的是单个开发者、“@组名”、“&别名”或*, 而选项值表示选项对路径所指向的目录的访问权限, r 表示读取权限, w 表示写入权限。*代表所有其他开发者。

在/etc/subversion/authz 文件添加如下代码:

```

[/]      //定义/home/svn 下所有的数据仓库
dranyu=rw //表示 dranyu 拥有对保存在/home/svn 目录下的数据仓库的读写权限

[groups]    //定义开发组
web3ds = dranyu,sunny //定义开发组 web3ds, 并使得其包含 dranyu 和 sunny 两个组员

[web3ds:/] //定义对数据仓库 web3ds 根目录的访问权限
@web3ds=rw //设置组 web3ds 下的所有组员对 web3ds 数据仓库的根目录有读写权限
sun=r      //sun 用户对 web3ds 数据仓库拥有只读权限
*=         //定义其他开发者不能访问 web3ds 项目

```

补充说明

在/etc/subversion/authz 文件出现的用户名, 需要在/etc/subversion/passwd 密码文件中设置用户的密码。而且编辑/etc/apache2/mods-available/dav_svn.conf 配置文件之后, 必须重启 apache2 服务器, 但编辑 authz 和 passwd 文件后, 并不需要重启 apache2 服务器。



2. 重启 apache2 服务器

完成对/etc/apache2/mods-available/dav_svn.cnf配置文件的配置和对/etc/subversion/authz文件的编辑后，重启 apache2 服务器。命令执行和结果显示如下：

```
root@dranyu-desktop:/etc/apache2/mods-enabled# /etc/init.d/apache2 restart //执行重启命令
* Restarting web server apache2
not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName
apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName
[ OK ]
```

3. 测试

测试 sun 对数据仓库的读写权限步骤如下。

(1) 使用 svn list 显示数据仓库 web3ds 的命令如下：

```
dranyu@dranyu-desktop:~/project/web3ds$ svn list --username sun //--username sun 表示以 sun 身份查看数据仓库 web3ds 的内容
认证领域: <http://localhost:80> web3ds subversion repository //用户验证的提示信息
“sun”的密码: //输入 sun 的密码
doc/
source/
thunk/
```

成功显示了 web3ds 根目录下的内容，结果显示了 web3ds 数据仓库的根目录包括了 doc、source 和 thunk 目录，也就说明了 sun 拥有对 web3ds 数据仓库的读取权限。

(2) 在本地副本 web3ds 目录下的 doc 文件夹里添加新文件 t.txt，执行命令和结果显示如下：

```
dranyu@dranyu-desktop:~/project/web3ds$ cd doc
dranyu@dranyu-desktop:~/project/web3ds/doc$ echo "This is Ok!" >t.txt
```

(3) 将 t.txt 添加到本地的 svn 记录里，命令执行和结果显示如下：

```
//svn add t.txt 命令表示将新文件 t.txt 注册到本地的 web3ds 项目中，并没提交到服务器
dranyu@dranyu-desktop:~/project/web3ds/doc$ svn add t.txt --username sun
A      t.txt //A 表示 t.txt 为新加文件
```

(4) 使用 svn commit 命令提交修改，执行命令和结果显示如下：

```
//注释： svn ci 命令表示将本地对 doc 目录的修改提交到服务器上，并且以 sun 身份提交
dranyu@dranyu-desktop:~/project/web3ds/doc$ svn ci --username sun
认证领域: <http://localhost:80> web3ds subversion repository
“sun”的密码: //输入 sun 的密码
svn: 提交失败(细节如下):
svn: 服务器发送了意外的返回值 (403 Forbidden)，在响应 “MKACTIVITY” 的请求
“/svn/web3ds!svn/act/11eb8078-a5ec-4062-b82d-60debaeb042e” 中
svn: 您的提交信息保留在临时文件中:
svn: '/home/dranyu/project/web3ds svn-commit.tmp'
```

结果显示表明 sun 不具有对数据仓库 web3ds 的写入权限。使用同样的方法，可测试 dranyu 用户和 sunny 用户对 web3ds 数据仓库的访问权限。

补充说明

svn 命令的使用在下一小节详述。



22.3 svn 字符界面客户端

svn 服务器配置好了数据仓库后，开发者就可使用数据仓库管理开发项目。管理 svn 的数据仓库可使用字符操作界面的方式也可使用图形操作界面的方式。安装 Subversion 软件包后，用户就可用户 svn 的相关命令完成版本控制的工作，包括检出文件、提交文件、更新文件和查看日志等。

22.3.1 检出文件

项目文件保存在 svn 服务器上，开发者需要将数据仓库的文件保存为一个在本地计算机的副本。将服务器上的指定版本号文件下载到本地计算机的操作，就叫文件的检出。使用 svn co 命令完成操作，命令的格式为：

```
svn co [本地机目录名] 数据仓库路径 [选项]
```

其中常用的选项如下：

有效选项如下。

- -r parameter：版本参数 parameter 可以是版本号，如 1 表示版本 1；也可以是日期时间，如{2009-12-15 20:23}表示此时间后的最早版本。若希望选择版本库中的最新版本，此参数 parameter 的值为 HEAD；若选择工作副本的基线版本，此参数 parameter 的值为 BASE。
- -q：此选项表示不打印信息，或只打印概要信息。

svn 命令的全局选项如下：

- --username parameter：此选项表示以指定的用户名为 parameter 连接 svn 服务器。
- --password parameter：指定用户的密码为 parameter。
- --no-auth-cache：此选项表示不要缓存用户认证令牌。
- --non-interactive：此选项表示不要交互提示。
- --config-dir DIR：此选项表示从目录 DIR 读取用户配置文件。

例 22-3：检出 web3ds 数据仓库

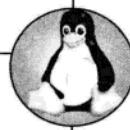
检出 web3ds 数据仓库到本地计算机，将副本保存到/home/dranyu/project 下，命令执行和结果显示如下：

```
dranyu@dranyu-desktop:~/project$ svn co http://localhost/svn/web3ds //检出命令
A   web3ds/trunk // A 表示添加，此行表示已在当前目录 project 下添加了 web3ds/trunk
...
A   web3ds/tags
取出版本 1。 //显示已检出数据仓库 web3ds 的第 1 个版本
```

此 svn co 命令的当前的工作目录 project 下创建一个文件目录 web3ds，保存 svn 服务器上的 web3ds 项目的文件。

例 22-4：检出 web3ds 数据仓库的第 1 个版本

将数据仓库 web3ds 第 1 版本上的 trunk 目录的内容检出到 web3ds1 目录下，执行命令和



结果显示如下：

```
//副本的名称指定为 web3ds1, “-r 1”表示获取第1个版本的文件  
dranyu@dranyu-desktop:~/project$ svn co -r 1 http://localhost/svn/web3ds/trunk web3ds1  
A   web3ds1/doc  
A   web3ds1/src  
取出版本 1。
```

22.3.2 添加文件

开发者在本地计算机工作副本添加了文件，在提交文件之前需要使用 `svn add` 命令往数据仓库中添加新的文件。`svn add` 命令的基本格式如下。

`svn add` 文件路径 [选项]

全局选项与 `svn co` 命令一样，其常用的选项如下。

- `--targets ARG`: 此选项指定传递文件 ARG 内容为附件参数。
- `-q` : 此选项表示不打印信息，或只打印概要信息。
- `--force`: 此选项表示强制操作运行。
- `--parents`: 此选项表示在操作过程中增加中间父目录。

例 22-5：向 web3ds 数据仓库添加新文件

进入到本地计算机的 web3ds 副本下的 trunk/doc 目录，创建一个名为 `readMe.txt` 文件，然后以 `dranyu` 的身份添加此文件，命令执行和结果显示如下：

```
dranyu@dranyu-desktop:~/project/web3ds/trunk/doc$ echo "This is web3ds project for 3D  
world" >readMe.txt //在本地的 doc 目录下添加 readMe.txt 文件  
dranyu@dranyu-desktop:~/project/web3ds/trunk/doc$ svn add readMe.txt --username  
dranyu  
A      readMe.txt //将 readMe.txt 添加到本地 svn 管理文件中，在下一次提交中添加  
到 svn 服务器上。
```

命令行中，`readMe.txt` 为所需要添加的文件名，“`--username dranyu`”是全局变量，表示以 `dranyu` 的身份执行操作。`svn add` 命令一次不单可添加一个文件，还可结合匹配符添加文件。如 `svn add r*` 表示添加以 `r` 开头的所有文件。

22.3.3 提交文件

本地计算机对项目所做的修改并没使 `svn` 服务器上的版本进行修改。将本地计算机的修改上传到服务器的操作就叫文件的提交。提交文件使用到 `svn ci` 命令，其基本格式如下：

`svn ci` [选项] [路径]

除了全局选项外，`svn ci` 命令常用选项如下：

- `-q` : 此选项表示不打印信息，或只打印概要信息。
- `--depth parameter`: 此选项表示受深度参数 parameter (“empty”, “files”, “immediates”，或“infinity”) 约束的操作。
- `--targets file`: 此选项指定传递文件 file 内容为附件参数。
- `--no-unlock`: 此选项表示不要解锁目标。
- `-m message`: 此选项指定日志信息为 message。
- `-F file`: 此选项表示从文件 file 读取日志信息。
- `--force-log`: 此选项表示强制校验日志信息资源。

- **--keep-changelists:** 此选项表示不要在提交后删除修改列表。

例 22-6: 提交本地副本的修改

将添加 readMe.txt 文件的操作提交到数据仓库中，并且以 dranyu 的身份提交，日志信息为“add readMe.txt file”，命令执行和结果显示如下所示：

```
dranyu@dranyu-desktop:~/project/web3ds/trunk/doc$ svn ci -m "add readMe.txt file."
--username dranyu
认证领域: <http://localhost:80> web3ds subversion repository
"dranyu" 的密码: //输入 dranyu 的 svn 密码
增加          doc/readMe.txt
传输文件数据 .
提交后的版本为 2.
```

命令行中，“-m”选项表示将后面的字符串作为提交的日志添加到 svn 系统的日志管理文件中，“--username dranyu”选项是 svn 命令的全局选项，表示以 dranyu 的身份执行操作。

补充说明

使用 svn 命令系列的命令时，若本地操作的当前目录在数据仓库的副本里，开发者可直接执行 svn 命令而不需设置数据仓库参数，如例 22-6 的提交命令没有设置 http:// localhost/ svn/web3ds/trunk/doc 路径。

22.3.4 更新文件

作为开发者在提交程序修改之前一定要执行更新文件的操作，以保持与服务器上的版本一致，而且也是为了解决在修改期间其他开发人员已做提交所引发的冲突问题。使用 svn update 命令更新本地副本，命令的基本格式如下：

```
svn update [选项] [路径]
```

除了全局选项，svn update 命令常用选项如下：

- **--accept ARG:** 此选项指定自动解决冲突动作的方式为 ARG，其中 ARG 的值包括 base、mine-full、theirs-full、edit。base 表示使用副本的基线，即系统原本的版本； mine-full 表示应用这一次的修改；theirs-full 表示应用第三者的修改；edit 表示若有冲突则启动编辑器，编辑冲突。

例 22-7: 更新 svn 服务器上的版本

进入本地计算机的 web3ds1 项目目录，更新 web3ds1 副本，命令执行和结果显示如下：

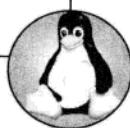
```
dranyu@dranyu-desktop:~/project/web3ds1$ svn update
A  doc/readMe.txt //在副本上添加 doc/readMe.txt 文件
更新到版本 2.
```

当前的工作目录是 web3ds 数据仓库工作副本的根目录，执行 svn update 命令就会执行更新 web3ds 工作副本的操作。

例 22-8: 更新到指定版本

在例 22-7 中，web3ds1 工作副本已更新到版本 2，现在将 web3ds1 工作副本更新到版本 1，命令执行和结果显示如下：

```
dranyu@dranyu-desktop:~/project/web3ds1$ svn update -r 1 //使用-r 1 选项，更新到指定的
版本 1
D  doc/readMe.txt //将版本 2 的 readMe.txt 文件删除
更新到版本 1
```



22.3.5 查看日志

日志对软件项目的开发来说十分重要。通过查看日志，可清楚地了解到项目的进度，查看开发者对程序的修改，可生成一个修改的说明文档。查看日志使用 `svn log` 命令，其命令基本格式如下：

```
svn log [路径] [选项]
```

除了 `svn` 的全局选项外，`svn log` 命令的常用选项如下。

- `-v`: 此选项表示打印附加信息。
- `-g`: 此选项表示从合并历史使用/显示额外信息。
- `-c ARG`: 此选项表示显示指定用户 ARG 所引起的改变。
- `--stop-on-copy`: 此选项表示查看不跨越不同工作副本的日志信息，即查看同一个副本连续提交的日志信息。
- `--xml`: 此选项表示将日志输出为 XML 文档。

例 22-9：查看当前副本原始版本基线的日志

查看 `web3ds` 数据仓库目前的日志信息，命令执行和结果显示如下：

```
dranyu@dranyu-desktop:~/project/web3ds$ svn log
-----
的开始, 结束标记
r2 | dranyu | 2009-12-12 16:57:46 +0800 (六, 2009-12-12) | 1 line
add readMe.txt file.                                //日志内容
-----
r1 | dranyu | 2009-12-12 10:30:54 +0800 (六, 2009-12-12) | 2 lines
project web3ds                                     //日志内容
-----
```

在日志开始结束标记下一行显示了日志来自的版本，如 `r2` 表示第 2 版本；执行修改操作的用户名，如 `dranyu`；修改的日期时间，如“`2009-12-12 16:57:46 +0800 (六, 2009-12-12)`”；日志内容的行数，如 `1 line` 表示 1 行日志内容。

例 22-10：查看日志信息时，显示附加信息

查看 `web3ds` 数据仓库日志信息，并显示被修改的文件信息，命令执行和结果显示如下：

```
dranyu@dranyu-desktop:~/project/web3ds$ svn log -v //使用-v 选项显示附加信息
-----
r2 | dranyu | 2009-12-12 16:57:46 +0800 (六, 2009-12-12) | 1 line
改变的路径:      //所做的修改
    A /trunk/doc/readMe.txt

add readMe.txt file.

-----
r1 | dranyu | 2009-12-12 10:30:54 +0800 (六, 2009-12-12) | 2 lines
改变的路径:      //所做的修改
    A /branches
    A /tags
    A /trunk
    A /trunk/doc
    A /trunk/src

project web3ds
-----
```

补充说明

svn 命令系列并不是以上几个命令，其中还包括：help、info、list、diff、delete 等子命令。使用命令“svn help [子命令]”可查看各子命令的详细用法。



22.4 svn 图形界面客户端 esvn

使用字符操作界面管理 svn 的数据仓库并不是一个程序员的首选。为了减小程序员在使用 svn 数据仓库时的麻烦，Ubuntu 集成了在 Linux 系统上使用广泛的 svn 图形界面客户端 esvn，并将其收入了 main 软件库中。使用 esvn 客户端，简化了 svn 的操作。

22.4.1 安装 esvn

打开新立得软件包管理器，搜索 esvn，然后选项 esvn 软件包并安装。或者在终端上输入如下命令进行安装：

```
root@dranyu-desktop:~# apt-get install esvn
```

安装 esvn 后，esvn 的应用程序保存在 /usr/bin/ 目录下，直接在终端上输入 esvn 即可启动 esvn 的图形操作界面如图 22-1 所示。

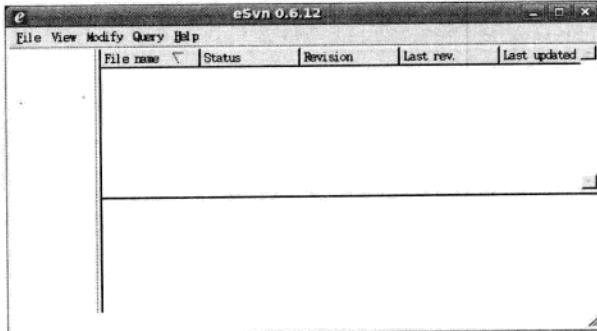


图 22-1 esvn 主界面

22.4.2 配置 esvn

在使用 esvn 之前，需要对它进行一番设置，如创建启动程序、设置显示字体和用户账号等。

1. 创建启动器

在桌面上为 esvn 创建一个启动器，方法如下：在桌面的快捷菜单中，选择“创建启动器”菜单命令，在弹出的“创建启动器”对话框中输入相关的信息，然后单击“确定”按钮，创建一个名为 esvn 的启动器，如图 22-2 所示。

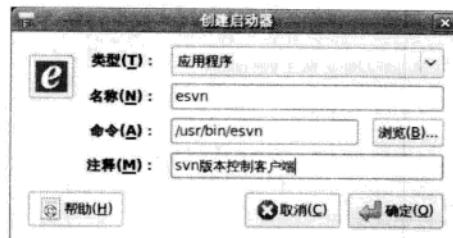


图 22-2 创建 esvn 启动器

创建了 esvn 客户端程序的启动器后，在桌面上出现 esvn 应用程序的图标，双击此图标即可快速启动 esvn 应用程序。

2. 设置显示的字体

esvn 默认使用的字符集是英文的，是为了防止在显示中文内容的时候出现乱码的问题。为了使用 esvn 中正常显示中文，需要对 esvn 的显示进行设置。

选择“File”|“Options”菜单命令，在弹出的“Options”对话框中，选择“Fonts”选项卡，单击各项后面的浏览按钮，选择“AR PL UMING CN”字体，操作结果如图 22-3 所示，然后按下“OK”按钮完成设置。

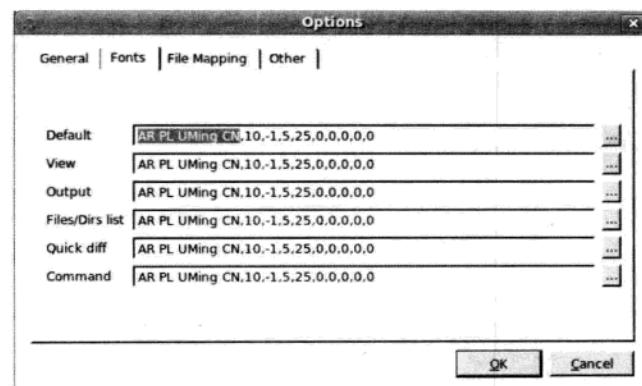


图 22-3 设置显示字体

22.4.3 检出文件

使用 esvn 检出文件的具体步骤如下：

- (1) 选择“File”|“Checkout”菜单命令，弹出“Checkout”对话框。
- (2) 在“Checkout”对话框中输入相关信息。在 URL 输入栏上输入 svn 服务器上数据仓库的路径，如 `http://localhost/svn/web3ds`；在 Username 输入栏上输入 svn 服务器的用户，如 dranyu；在 Password 输入栏中输入用户的密码；在 Local Path 输入栏上输入或选择保存项目副本的路径，如`/home/dranyu/project/`。其他参数保持默认，如图 22-4 所示。
- (3) 设置完成之后，单击“OK”按钮，esvn 开始从 svn 服务器上下载 web3ds 数据仓库到本地计算机的指定位置`/home/dranyu/project`，执行命令结果显示如图 22-5 所示。

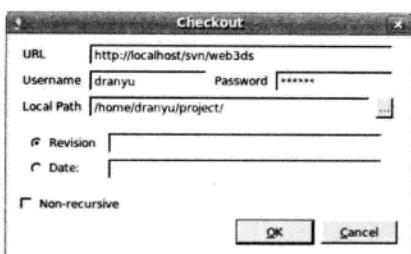


图 22-4 检出文件

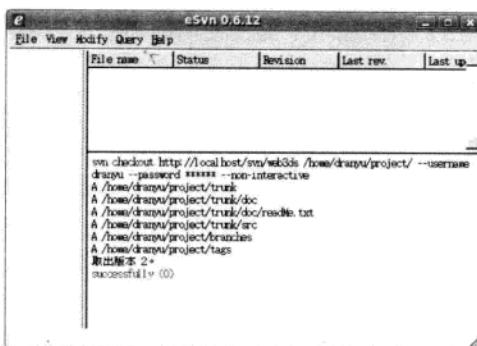


图 22-5 检出文件的结果显示

22.4.4 打开工作副本

数据仓库的文件检出到本地计算机后，没有在 eSvn 的操作界面上打开，所以检出后，打开保存在本地的工作副本的步骤为：选择“File”|“Open working directory”菜单命令，从弹出的 eSvn 对话框中，选择“/home/dranyu/project/”目录打开工作副本。

22.4.5 提交文件

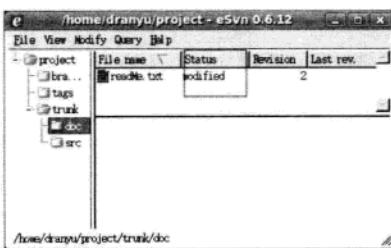


图 22-6 修改后的文件状态

文件只有在修改之后才具有提交文件的意义。为讲解 eSvn 提交文件的功能，需要使用 Gedit 文本编辑器，编辑/home/dranyu/project/trunk/doc/readMe.txt 文件。修改文件 readMe.txt 后，双击桌面上的启动器，启动 eSvn 应用程序。在左边的工作副本目录上选择“trunk/doc”目录选项，结果显示如图 22-6 所示。

结果显示目录 doc 上的 readMe.txt 的 Status 为 modified，表示此时 readMe.txt 文件在本地上已被修改。

选择“Modify”|“Commit”菜单命令，弹出 Commit 对话框，在中间的输入框中输入日志信息，如图 22-7 所示。完成提交后，结果显示如图 22-8 所示。

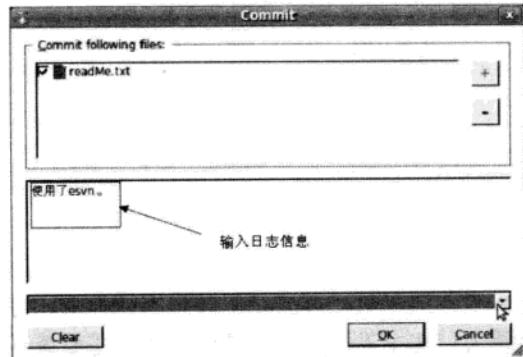


图 22-7 提交文件

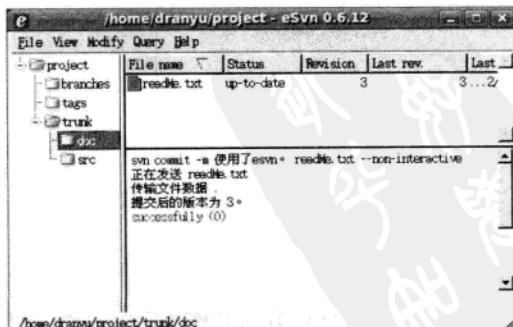


图 22-8 结果显示



图 22-8 结果显示了提交使用的命令，以及执行命令的结果信息。

22.4.6 更新工作副本

esvn 客户端默认提供了两个更新副本的菜单包括“update”和“update revision”。

第一种方式的操作为：在主界面的目录列表中选中需要更新的副本或副本目录，然后选择“Modify”|“update”菜单命令，即可完成更新到最新的版本。

第二种方式的操作为：在主界面的目录列表中选中需要更新的副本或副本目录，然后选择“Modify”|“update revision”菜单命令，弹出 Update 对话框，在对话框中选中 Revision 选项，在输入栏上输入版本号，如 2，表示将当前的副本更新到版本 2，如图 22-9 所示；或选择 Date 选项，输入一个日期，如“2009-12-12 23:20”，将当前副本更新到指定日期时间的版本，如图 22-10 所示，然后再单击“OK”按钮，完成操作。

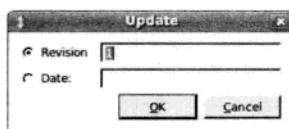


图 22-9 指定版本号更新

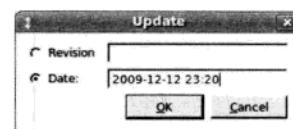


图 22-10 指定日期时间更新

使用以上两种方式更新副本的结果显示如图 22-11 所示。

```
svn update trunk --non-interactive
U trunk/doc/readme.txt
更新到版本 3。
successfully (0)
svn update trunk --non-interactive
版本 3。
successfully (0)
svn update trunk --non-interactive
版本 3。
successfully (0)
svn update -r 2 trunk --non-interactive
U trunk/doc/readme.txt
更新到版本 2。
successfully (0)
svn update -r (2009-12-12 23:20) trunk --non-interactive
版本 2。
successfully (0)
```

图 22-11 更新结果

22.4.7 查看日志

esvn 客户端提供了日志查看器工具，使用日志查看器工具，可清楚了解日志的信息，可快速搜索日志。查看数据仓库的日志文件一般步骤为：在主界面左边目录树上选中需要查看日志的目录，如 trunk，然后选择“Query”|“Log”菜单命令，弹出相应的日志对话框，如

trunk 对话框，如图 22-12 所示，再选择日志列表中的记录，就可在日志内容框中查看相应的日志内容。

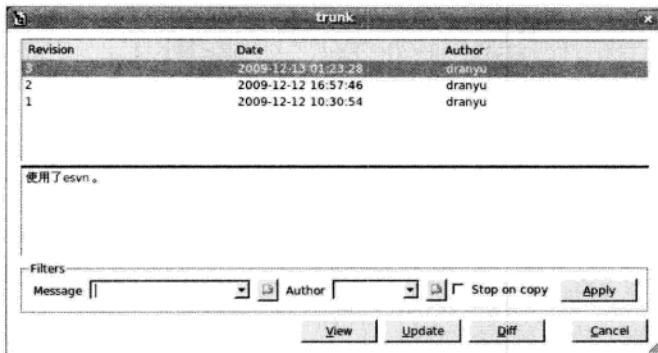


图 22-12 查看日志

用户还可设置 Filters(过滤器)选项框中的参数，搜索相关的日志信息。若在 Message 输入栏输入字符串，如 esvn，按下“Apply”按钮后日志列表显示日志内容与 esvn 字符串匹配的日志记录；若在 Author 中输入用户名，如 dranyu，按下“Apply”按钮后日志列表就会显示 dranyu 所做的所有日志记录；若这两个都输入了内容，按下“Apply”按钮后列表显示的日志记录需要符合这两个条件。

22.5 小结



使用版本控制软件，可方便管理程序开发过程中所产生的程序代码和项目文档，让项目负责人把握项目的进度，可跟踪项目开发的整个流程。本章主要讲述了如何构建 svn 服务器，配置使用 apache2 与 svn 的结合。在 svn 服务器配置好，使用 svn 命令导入项目数据、检出文件、添加文件、提交文件和查看修改日志等。最后讲述了在 Ubuntu 中如何使用 svn 的图形操作界面客户端——esvn 应用程序。

22.6 习题



1. 安装 Subversion 软件包和 libapache-svn 软件包。
2. 创建一个名为 web3ds 的数据仓库，并使用 svn import 命令将一个 web3ds 项目导入到 web3ds 数据仓库。
3. 配置使用 apache2 的 Web 服务器访问 svn 服务器上的数据仓库 web3ds，并使用 htpasswd 命令添加 passwd 用户密码文件，设置 svn 服务器中 web3ds 数据仓库的用户名为 dranyu，以及用户 dranyu 的密码为 123456。



4. 使用 htpasswd 命令添加 tom 和 sunny 用户，然后使用文本编辑器编辑 authz 用户权限文件，为 svn 服务器添加一个 web3ds 项目组，组员包括 dranyu、tom 和 sunny。
5. 使用 svn co 命令为 web3ds 数据仓库在本地计算机上创建一个工作副本，副本名称为 web3ds。
6. 在 web3ds 副本目录下添加文件 readme.txt，然后使用 svn add 将 readme.txt 文件添加到数据仓库中，再使用 svn ci 命令将本地的修改提交到 svn 服务器上。
7. 修改 web3ds 副本上 readme.txt 文件的内容，然后使用 svn ci 命令将修改提交到 svn 服务器上。
8. 使用 svn log 命令查看 web3ds 数据仓库的修改日志；查看用户 dranyu 所写的修改日志。
9. 安装 esvn 软件包，设置 esvn 的相关参数，并为 esvn 应用程序在桌面上创建一个启动器 esvn。
10. 在 esvn 应用程序上，完成数据仓库 web3ds 的检出操作。
11. 使用 esvn 应用程序打开 web3ds 工作副本，然后完成提交、更新和查看日志等操作。



[General Information]

书名 = UBUNTU LINUX从初学到精通

作者 = 华清远见嵌入式培训中心 , 潘光洋等编著

页数 = 478

SS号 = 12767144

出版日期 = 2011.04

出版社 = 电子工业出版社

SLIB-JPG=http://image4.5read.com/image/ss2jpg.dll?did=n27
&pid=F6D703627F8E759EDE85DD97C4AB9F912A40BDFD367BABD484434
665465635A6B1E1E2826DD8DB810321B5587A0B1B653D50C84DC5584D4
EDC00D1E4D833261D0A5936AE83852C12E4B576558B553CCFF3251C384
F7D3304701C8913EA5F991C61B7C68878B48260FDBDD5723383BFDB068
7&jid= /