

Linux · Qt



清华大学出版社
<http://www.tup.com.cn>



北京交通大学出版社
<http://press.bjtu.edu.cn>

Linux与Qt

程序设计

陈 爽 主 编

秦攀科 贺 荣 杨 晨 副主编



Linux 与 Qt 程序设计

主 编 陈 爽

副主编 秦攀科 贺 荣 杨 晨

清 华 大 学 出 版 社

北京交通大学出版社

· 北京 ·

内 容 简 介

本书从四个方面介绍 Linux 系统与 Qt 程序设计的知识：第一方面介绍 Linux 基础知识，包括发展历史、常用的命令、文件系统等；第二方面主要介绍 Qt 的基础知识，包括如何编写第一个 Qt 程序、信号与槽、Qt 事件机制及 MVC 模式等；第三部分着重介绍 Qt 界面设计的核心内容，包括常用控件、布局管理、对话框及主窗体等；第四部分通过完整的实例介绍 Qt 的典型应用。

本书适用于具有一定 C/C++ 程序基础的开发人员阅读，也可以作为 Qt 开发过程中的工具参考书。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13501256678 13801310933

图书在版编目 (CIP) 数据

Linux 与 Qt 程序设计 / 陈爽主编. — 北京：清华大学出版社；北京交通大学出版社，2011.12

ISBN 978-7-5121-0812-7

I. ①L… II. ①陈… III. ①Linux 操作系统-程序设计 ②软件工具-程序设计
IV. ①TP316.89 ②TP311.56

中国版本图书馆 CIP 数据核字 (2011) 第 253186 号

责任编辑：郭东青

出版发行：清华大学出版社 邮编：100084 电话：010-62776969 <http://www.tup.com.cn>

北京交通大学出版社 邮编：100044 电话：010-51686414 <http://press.bjtu.edu.cn>

印刷者：北京瑞达方舟印务有限公司

经 销：全国新华书店

开 本：185×260 印张：22 字数：549 千字

版 次：2011 年 12 月第 1 版 2011 年 12 月第 1 次印刷

书 号：ISBN 978-7-5121-0812-7/TP·675

印 数：1~4 000 册 定价：33.00 元

本书如有质量问题，请向北京交通大学出版社质监组反映。对您的意见和批评，我们表示欢迎和感谢。

投诉电话：010-51686043，51686008；传真：010-62225406；E-mail: press@bjtu.edu.cn。

前 言

本书从“Linux 基础知识”、“Qt 程序基础”、“Qt 界面设计”和“进阶应用”四个方面介绍 Linux 系统与 Qt 程序设计，使读者能够初步了解 Linux 操作系统的一般特性并初步掌握 Linux 下 Qt 程序开发的过程。

Linux 基础知识部分介绍了 Linux 的发展历史，Ubuntu 的安装与一些设置、一些 Linux 下的常用命令及 Ubuntu 下开发工具的设置，最后介绍 Linux 文件系统的基础知识。

Qt 程序基础部分首先介绍了如何通过 Qt Creator 的下载与安装，如何编写第一个程序，如何调试、编译一个 Qt 应用程序及 Qt SDK 的基础知识，并介绍了 Qt 的通用编程方法、信号与槽使用、Qt 事件机制及 MVC 模型与视图的基础知识。

Qt 界面设计部分通过介绍 Qt 常用控件、Qt 的布局管理和对话框等，最后通过一个完整的 Qt QMainWindow 主窗口应用程序使读者了解 Qt 界面设计一般知识。

Qt 进阶应用部分介绍了 Qt 通用类，Qt 的文件操作包括文件、目录的操作和典型的 XML 操作应用、Qt 的多线程开发、Qt 的数据库编程、Qt 的网络编程及如何使用使用 Qt 绘图等几个部分。读者学习掌握后可以编写较为复杂的 Qt 应用程序。

近年来作者因为工作需要越来越多的开始接触 Linux 下的应用程序开发，但是一直苦恼于 Linux 下缺乏 Windows 下良好的可视化开发环境。当接触到 Qt 以后，对这一印象有了大大的改观。古语有云：工欲善其事必先利其器。一个优秀的开发工具，对于实际项目的推进是非常有用的。也借此机会对过去的学习做一个简要的总结。

本书的代码在 Ubuntu 10.10 和 Ubuntu 11.04 下测试通过。

由于完成时间仓促，以及个人理解可能不同，本书中难免有不足，恳请读者批评指正，如果发现问题和批评建议，请发邮件到 chenshuangnwpu@126.com。

感谢一起合作的朋友，是我们的共同努力才使该书得以完成。

编 者

2010 年 12 月

目 录

第一部分 Linux 基础知识

第 1 章 Linux 系统概述	1	3.1.2 ls	28
1.1 Linux 简介	1	3.1.3 cd	30
1.2 Linux 内核基本概念	2	3.1.4 clear	31
1.2.1 Linux 内核的主要内部子系统	3	3.1.5 mkdir	31
1.2.2 Linux 内核命名方法	5	3.1.6 rmdir	32
1.3 Linux 桌面环境	5	3.1.7 find	32
1.3.1 KDE 简介	5	3.1.8 cp	34
1.3.2 GNOME 简介	6	3.1.9 rm	35
1.4 Linux 的发行版本	7	3.1.10 chmod	36
1.4.1 Fedora	7	3.1.11 locate	37
1.4.2 Debian	8	3.2 文件编辑	38
1.4.3 Ubuntu	9	3.2.1 wc	38
1.4.4 Red Hat	9	3.2.2 cat	39
1.4.5 openSUSE	10	3.2.3 ed	40
1.4.6 红旗 Linux	11	3.2.4 sort	41
1.5 Linux 下的软件安装	11	3.2.5 look	41
1.5.1 Redhat Package Manager	11	3.2.6 comm	42
1.5.2 dpkg	11	3.2.7 expr	43
第 2 章 Ubuntu 的安装与开发		3.3 网络命令	44
环境设置	12	3.3.1 ifconfig	44
2.1 Ubuntu 的安装	12	3.3.2 ping	45
2.2 Ubuntu 常用设置	14	3.3.3 telnet	46
2.2.1 新建用户	14	3.3.4 netstat	46
2.2.2 网络设置	15	3.3.5 traceroute	47
2.3 Ubuntu 的版本	16	3.3.6 route	48
2.4 开发环境的设置	18	3.3.7 ip	48
2.4.1 Ubuntu 下的 Java 安装与配置	18	3.3.8 host	49
2.4.2 Ubuntu 下的 Tomcat 安装与配置	22	3.3.9 ufw	50
2.4.3 Ubuntu 下的 MySQL 安装		3.3.10 wget	51
与配置	24	3.3.11 ps	52
第 3 章 Linux 常用命令与配置	28	3.3.12 pgrep	53
3.1 文件与磁盘操作	28	3.3.13 kill	53
3.1.1 pwd	28	3.3.14 sudo	54

3.4 其他常用命令	54	4.2 Linux 的文件	61
3.4.1 用户管理相关操作	54	4.2.1 文件的结构	61
3.4.2 退出控制台	57	4.2.2 文件的属性	61
3.4.3 apt-get 安装软件	57	4.2.3 可执行文件	62
第4章 Linux 文件系统	58	4.2.4 文件链接	62
4.1 Linux 分区	58	4.3 Linux 的目录	62
4.1.1 磁盘设备	58	4.3.1 工作目录与用户主目录	63
4.1.2 挂载点	59	4.3.2 路径与通配符	63
4.1.3 Ext4	59	4.3.3 Linux 目录结构	63
4.1.4 交换空间	60		

第二部分 Qt 程序基础

第5章 Qt Creator 下载与安装	65	7.2 程序资源与管理	85
5.1 Qt Creator 下载	65	7.2.1 界面交互文件 (.ui)	85
5.1.1 使用软件中心下载 Qt Creator	65	7.2.2 程序资源文件 (.qrc)	87
5.1.2 访问 Qt 网站下载相关资源	66	7.3 应用程序的创建	89
5.2 第一个 Qt 程序	66	7.3.1 创建 GUI 程序	89
5.3 Qt Creator 介绍	70	7.3.2 创建库工程	93
5.3.1 Qt Creator 界面介绍	70	7.3.3 创建 console 程序	96
5.3.2 编译、调试、发布设置	73	第8章 信号与槽	100
5.4 Qt Designer 介绍	74	8.1 概述	100
第6章 Qt 基础	75	8.1.1 信号	100
6.1 Qt 概述与特性	75	8.1.2 槽	101
6.1.1 Qt 简介	75	8.2 信号与槽的关联	101
6.1.2 Qt 特性	76	8.2.1 单信号与槽的实例	102
6.2 Qt 与 X11 的关系	77	8.2.2 多信号与槽的实例	105
6.2.1 什么是 X11	77	8.3 元对象工具	106
6.2.2 Qt 与 X11	78	8.4 应注意的问题	106
6.3 Qt 与 KDE 的关系	78	第9章 Qt 事件	110
6.3.1 KDE 简介	78	9.1 Qt 事件机制	110
6.3.2 Qt 与 KDE	79	9.1.1 什么是 Qt 事件	110
6.4 Qt 库和组件	79	9.1.2 Qt 事件的响应	111
6.4.1 Qt SDK 简介	79	9.1.3 自定义 Qt 事件	116
6.4.2 Qt 主要组件介绍	81	9.2 Qt 事件的过滤	117
第7章 Qt 编程方法	82	9.2.1 Qt 事件的派发	117
7.1 Qt 应用程序的基本结构	82	9.2.2 安装事件过滤器	118
7.1.1 Qt 工程文件	82	9.3 Qt 事件的截获	119
7.1.2 头文件和源文件	84	9.3.1 截获 win32 窗口消息	120
7.1.3 元对象文件	84	9.3.2 截获 x11 窗口事件	123

9.4 Qt 按键事件实例	127	10.3.1 QcolumnView 类	133
第 10 章 MVC 模型/视图	131	10.3.2 QListView 类	135
10.1 Qt 的 Model/View 架构	131	10.3.3 QTableView 类	137
10.2 Qt 中 Model/View 相关类	132	10.3.4 QTreeView 类	139
10.3 典型实例	133		

第三部分 Qt 界面设计

第 11 章 Qt 常用控件	144	12.2.3 QGridLayout	172
11.1 按钮控件	144	12.2.4 QFormLayout	173
11.1.1 QPushButton 类	144	12.2.5 QStackedLayout	173
11.1.2 QButtonGroup 类和 QRadioButton 类	145	12.3 综合实例	174
11.2 文本类控件	147	第 13 章 对话框	178
11.2.1 QLineEdit 类	147	13.1 QDialog 类	178
11.2.2 QTextEdit 类	149	13.2 子类化 QDialog 类	178
11.3 时间类控件	151	13.3 常用标准对话框	182
11.3.1 QTime 类和 QTimeEdit 类	151	13.3.1 QColorDialog 类	182
11.3.2 QDate 类和 QDateEdit 类	153	13.3.2 QFileDialog 类	184
11.3.3 QCalendarWidget 类	155	13.3.3 QFontDialog 类	187
11.4 其他控件	156	13.3.4 QMessageBox 类	189
11.4.1 QComboBox 类	156	13.3.5 QProgressDialog 类	192
11.4.2 QLCDNumber 类	158	第 14 章 QMainWindow 主窗口 应用程序	193
11.4.3 QProgressBar 类	159	14.1 简介	193
11.4.4 QSpinBox 类和 QDoubleSpinBox 类	160	14.2 菜单栏、工具栏与状态栏	194
11.4.5 QSlider 类	163	14.2.1 菜单栏	194
11.4.6 QTreeWidget 类	165	14.2.2 工具栏	195
11.4.7 QTabWidget 类	168	14.2.3 状态栏	196
第 12 章 布局管理	170	14.3 工作区	196
12.1 Qt 布局管理介绍	170	14.4 滚动视图与锚接窗口	197
12.2 布局管理器	170	14.4.1 滚动视图	197
12.2.1 QHBoxLayout	171	14.4.2 锚接窗口	197
12.2.2 QVBoxLayout	172	14.5 完整代码	198

第四部分 进阶应用

第 15 章 Qt 容器类	206	15.1.3 访问元素	210
15.1 通用顺序容器	206	15.1.4 查找与替换	214
15.1.1 类摘要	207	15.1.5 综合应用	215
15.1.2 常用操作函数	208	15.1.6 隐含共享	216

15.2 迭代器	218	18.2 连接数据库	264
15.2.1 Java 风格迭代器	218	18.2.1 MySQL 数据库连接	264
15.2.2 STL 风格迭代器	220	18.2.2 SQLite 数据库连接	265
15.2.3 STL 风格迭代器与通用算法	222	18.3 数据库基本操作	265
15.2.4 综合应用	226	18.3.1 添加	265
15.3 通用链式容器	228	18.3.2 浏览查询	266
15.3.1 类摘要	228	18.3.3 删除	266
15.3.2 常用操作函数	229	18.3.4 修改	267
15.3.3 访问元素	231	18.4 SQLite 数据库实例	267
15.4 关联容器	231	18.4.1 界面设计	267
15.4.1 类摘要	231	18.4.2 创建绑定数据库	269
15.4.2 常用操作函数	232	18.4.3 主函数与程序主界面	270
15.4.3 访问元素	233	18.4.4 数据库操作	273
第 16 章 文件操作	236	第 19 章 QT 网络编程	282
16.1 QFile 类	236	19.1 获取主机网络信息	282
16.1.1 读写二进制流文件	237	19.2 基于 UDP 和 TCP 的 聊天室	285
16.1.2 读写文本	238	19.2.1 基于 UDP 的聊天功能	286
16.2 目录操作	239	19.2.2 基于 TCP 的文件传输	289
16.3 文件管理	240	19.3 基于 Socket 的 Qt 与 Flex 通信	308
16.3.1 获取文件属性	240	19.4 HTTP 文件下载	314
16.3.2 监视文件系统变化	243	19.5 FTP 客户端	318
16.4 XML 文件操作	244	第 20 章 Qt 绘图	328
16.4.1 读取 XML 文件	244	20.1 QPainter 绘图	328
16.4.2 写入 XML 文件	249	20.1.1 QPen 类	329
第 17 章 Qt 多线程	251	20.1.2 QBrush 类	332
17.1 在 Qt 中创建一个线程	251	20.1.3 QFont 类	333
17.2 Qt 的同步控制	253	20.2 渐变填充绘图	335
17.2.1 QMutex 类	253	20.3 绘图与绘图设备	339
17.2.2 QReadWriteLock 类	256	20.4 坐标系统与坐标变换	340
17.2.3 QSemaphore 类	259	20.4.1 坐标系统	340
17.2.4 QWaitCondition 类	260	20.4.2 坐标系变换	341
17.3 使用线程池	262	20.4.3 坐标系状态的保存与恢复	344
第 18 章 Qt 数据库编程	264		
18.1 Qt 数据库开发基础	264		

第一部分 Linux 基础知识

第 1 章 Linux 系统概述

Linux 是一套免费使用和自由传播的操作系统，它主要用于基于 X86 系列 CPU 的计算机上。这个系统是由世界各地的成千上万的程序员设计和实现的，其目的是建立不受任何商品化软件的版权制约的、全世界都能自由使用的操作系统。通过本章的介绍，读者将会初步了解 Linux 的基础知识。

1.1 Linux 简介

Linux 是由芬兰人 Linus Torvalds 开发，并在 1991 年 10 月 5 日正式对外发布。由于具有结构清晰、功能简洁等特点，许多高等院校的学生和科研机构的研究人员纷纷把它作为学习和研究的对象。Linus Torvalds 从一开始，就决定自由扩散 Linux，包括源代码，他把源代码发布在网上，随即就引起编程高手和业余计算机爱好者的注意，他们通过互联网也加入了 Linux 的内核开发工作，一大批高水平程序员的加入，在更正原有 Linux 版本中错误的同时，也不断地为 Linux 增加新的功能，使得 Linux 迅猛发展。于是 Linux 快速从一个个人项目进化成为一个全球数千人参与的开发项目，正是因为 Linux 是采用 GPL (GNU General Public License)，在 GPL 保护之下，Linux 内核可以防止被商业使用，并且它还从 GNU 项目 (Richard Stallman 开发，其源代码要比 Linux 内核大得多) 的用户空间开发受益。

起初，Linux 操作系统并没有被称做 Linux，Linus 给他的操作系统取名为 FREAX，其英文含义是怪诞的、怪物、异想天开等意思。在他将新的操作系统上传到 ftp.funet.fi 服务器上时，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吧，于是 Linux 这个名称就开始流传下来。

Linux 系统的系统核心由起初的 Linux 0.01 发展为新的 Linux 2.60 版。它已经在全球获得了极大的成功，尤其在服务器操作系统的应用上，Linux 已经占据了优势地位。如今，Linux 已经成为一个稳定可靠、功能完善、性能卓越的操作系统。

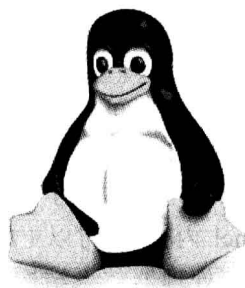


图 1-1 Linux 的标志

图 1-1 是 Linux 的标志企鹅 Tux。而将企鹅作为 Linux 标志是由 Linus Torvalds 提出的。“Tux”这个名字来源于 Torvalds UNIX。这个企



鹅图案在 1996 年的最佳 Linux 图标竞赛中被选中。现在 Tux 已经成为 Linux 和开源社群的标志。

Linux 操作系统和 UNIX 系统有很多相似之处，它也可以称为是 UNIX 操作系统的一种克隆系统。UNIX 操作系统是美国贝尔实验室的 Ken Thompson 和 Dennis Ritchie 于 1969 年夏在 DEC PDP-7 小型计算机上开发的一个分时操作系统。

1973 年 Ken Thompson 和 Dennis Ritchie 发明了 C 语言之后，着手将 UNIX 程序用高阶的 C 语言改写，这个时候已经是第四版了。C 的前身叫 B，也是他们两个人所发明的，为了改写 UNIX，他们将 B 语言做了一番改进，而成了 C 语言。这个划时代的动作让 UNIX 能够很轻易地被移植到各种不同的机器上，只要修改少许的原始程序代码，整个 UNIX 可以在很短时间内完成移植的动作。

1974 年，他们在 *Communications of the ACM* 正式发表了 UNIX，吸引了学术界的注意与参与。从此 UNIX 逐渐成为网络工作站的标准，让使用不同机器的使用者，都能使用类似的 UNIX 操作系统。

可以通过 http://en.wikipedia.org/wiki/File:Unix_history-simple.svg 看到今天各种各样的 UNIX 发行版。

虽然很多时候说 Linux 是基于 UNIX 的，这是因为这两者的基本思想十分相近。但 Linux 和 UNIX 的最大区别是，前者是开放源代码的自由软件，而后者是对源代码实行知识产权保护的传统商业软件。这应该是它们最大的不同，这种不同体现在用户对前者有很高的自主权，而对后者却只能去被动的适应；这种不同还表现在前者的开发是处在一个完全开放的环境之中，而後者的开发完全是处在一个黑箱之中，只有相关的开发人员才能够接触到产品的原型。

1.2 Linux 内核基本概念

在 Linux 操作系统中，一个最重要的概念就是内核 Kernel。图 1-2 显示了 Linux Kernel 在操作系统这一层次中的地位。

其中用户（或应用程序）空间是用户应用程序执行的地方。GNU C Library (glibc) 也在这里。它提供了连接内核的系统调用接口，还提供了在用户空间应用程序和内核之间进行转换的机制。

用户空间之下是内核空间，Linux 内核正是位于这里。内核和用户空间的应用程序使用的是不同的保护地址空间。每个用户空间的进程都使用自己的虚拟地址空间，而内核则占用单独的地址空间。

在图 1-2 中，Linux 内核可以进一步划分成三层。最上面是系统调用接口（System Call Interface），它实现了一些基本的系统功能，例如内存读写。系统调用接口之下是内核代码（Kernel 内核），可以更精确地定义为独立于体系结构的内核代码。这些代码是 Linux 所支持的所有处理器体系结构所通用的。在这些代码之下是依赖于体系结构的代码（Architecture-dependent kernel code），构成了通常称为 BSP（Board Support Package）的部分。这些代码用于特定体系结构的处理器和特定平台的代码。

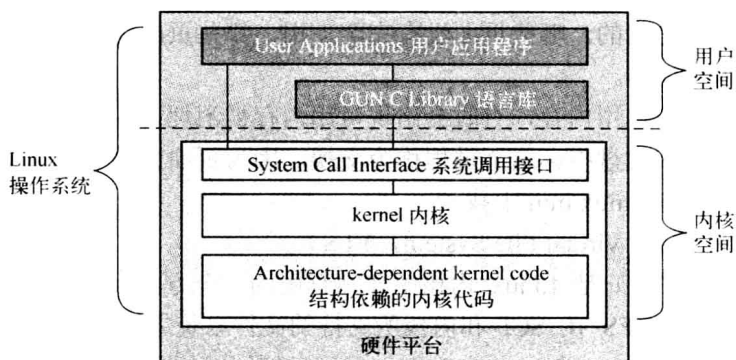


图 1-2 Linux Kernel 层次结构图

1.2.1 Linux 内核的主要内部子系统

图 1-3 显示了 Linux 内核的主要内部子系统。

包括以下几个主要模块。

(1) 系统调用接口（System Call Interface, SCI）

SCI 层提供了某些机制执行从用户空间到内核的函数调用。SCI 的接口依赖于体系结构，甚至在相同的处理器家族内也是如此。SCI 实际上是一个非常有用的函数调用多路复用和多路分解服务。在 `./linux/kernel` 中可以找到 SCI 的实现，并在 `./linux/arch` 中找到依赖于体系结构的部分。

(2) 进程管理（Process Management, PM）

进程管理的重点是进程的执行。在内核中，这些进程称为线程，代表了单独的处理器虚拟化（线程代码、数据、堆栈和 CPU 寄存器）。Linux 内核中并没有区分进程和线程两个程序设计中常用的概念。内核通过 SCI 提供了一个应用程序编程接口（API）来创建一个新进程（`fork`、`exec` 或 `Portable Operating System Interface[POSIX]` 函数），停止进程（`kill`、`exit`），并在它们之间进行通信和同步（`signal` 或者 `POSIX` 机制）。

进程管理还包括处理活动进程之间共享 CPU 的需求。内核实现了一种新型的调度算法，不管有多少个线程在竞争 CPU，这种算法都可以在固定时间内进行操作。这种算法就称为 `O(1)` 调度程序，它调度多个线程所使用的时间和调度一个线程所使用的时间是相同的。`O(1)` 调度程序也可以支持多处理器（称为对称多处理器或 `SMP`）。可以在 `./linux/kernel` 中找到进程管理的源代码，在 `./linux/arch` 中可以找到依赖于体系结构的源代码。

(3) 内存管理（Memory Management, MM）

操作系统内核所管理的另外一个重要资源是内存。Linux 包括了管理可用内存的方式，以及物理和虚拟映射所使用的硬件机制。

Linux 提供了对 4KB 缓冲区的抽象，用于兼容以分页方式管理的硬件，例如 `slab` 分配器。这种内存管理模式使用 4KB 缓冲区为基数，然后从中分配结构，并跟踪内存页使用情况。

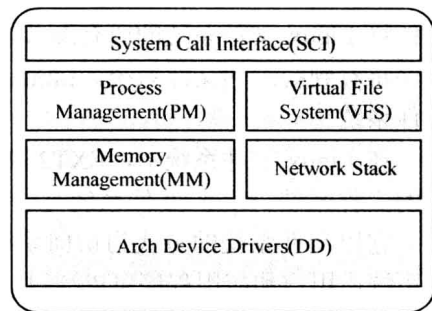


图 1-3 Linux 内核内部构造图



况，比如哪些内存页是满的，哪些页面没有完全使用，哪些页面为空。这样就允许根据系统需要来动态调整内存使用。

为了支持多个用户使用内存，有时会出现可用内存被消耗光的情况。由于这个原因，页面可以移出内存并放入磁盘中。这个过程称为交换，因为页面会被从内存交换到硬盘上。内存管理的源代码可以在 `./linux/mm` 中找到。

(4) 虚拟文件系统 (Virtual File System, VFS)

虚拟文件系统 (VFS) 是 Linux 内核中非常有用的一个方面，因为它为文件系统提供了一个通用的接口抽象。VFS 在 SCI 和内核所支持的文件系统之间提供了一个交换层。Linux 在 VFS 之上实现了对 `open`、`close`、`read` 和 `write` 等函数的通用 API 抽象。而在 VFS 下面是文件系统抽象，它定义了上层函数的实现方式。可以在 `./linux/fs` 中找到文件系统的源代码。文件系统层之下是缓冲区缓存，它为文件系统层提供了一个通用函数集（与具体文件系统无关）。这个缓存层通过将数据保留一段时间（或者预先读取数据以便在需要时就可用）优化了对物理设备的访问。缓冲区缓存之下是设备驱动程序，它实现了特定物理设备的接口。

VFS 使得 Linux 可以支持多个不同的文件系统，每个表示一个 VFS 的通用接口。由于软件将 Linux 文件系统的所有细节进行了转换，所以 Linux 核心的其他部分及系统中运行的程序将看到统一的文件系统。Linux 的虚拟文件系统允许用户同时能透明地安装许多不同的文件系统。

在 Linux 文件系统中，EXT2 文件系统、虚拟文件系统、`/proc` 文件系统是三个具有代表性的文件系统。`/pro` 文件系统是一类特殊的文件系统，只存在内存当中，而不占用外存空间。它以文件系统的方式为访问系统内核数据的操作提供接口。`/proc` 文件系统是一个伪文件系统，用户和应用程序可以通过 `/proc` 得到系统的信息，并可以改变内核的某些参数。

(5) 网络堆栈 (NetWork Stack)

网络堆栈在设计上遵循模拟协议本身的分层体系结构。TCP 等网络层协议是通过 SCI 进行调用的。`socket` 层是网络子系统的标准 API，它为各种网络协议提供了一个用户接口。从原始帧访问到 IP 协议数据单元 (PDU)，再到 TCP 和 User Datagram Protocol (UDP)，`socket` 层提供了一种标准化的方法来管理连接，并在各个终点之间移动数据。可以在 `./linux/net` 中找到内核中的网络源代码。

(6) 设备驱动程序 (Device Driver, DD)

Linux 内核中有大量代码都在设备驱动程序中，它们能够运转特定的硬件设备。Linux 源码树提供了一个驱动程序子目录，这个目录又进一步划分为各种支持设备，例如 Bluetooth、I2C、serial 等。可以在 `./linux/drivers` 中找到相关的代码。

(7) 依赖体系结构的代码 (Arch)

尽管 Linux 很大程度上独立于所运行的体系结构，但是有些元素则必须考虑体系结构才能正常操作并更高效地实现。`./linux/arch` 子目录定义了内核源代码中依赖于体系结构的部分，其中包含了各种特定于体系结构的子目录（共同组成了 BSP）。对于一个典型的桌面系统来说，使用的是 `i386` 目录。每个体系结构子目录都包含了很多其他子目录，每个子目录都关注内核中的一个特定方面，例如引导、内核、内存管理等。这些依赖于体系结构的代码可以在 `./linux/arch` 中找到。



1.2.2 Linux 内核命名方法

Linux 内核的版本号是有一定的规则的，版本号遵从的格式通常是：

主版本号.次版本号.修正号。

主版本号和次版本号标志着重要的功能变动；修正号表示较小的功能变动。以 2.6.12 版本为例，2 代表主版本号，6 代表次版本号，12 代表修正号。其中次版本号还有特定的意义：如果次版本号是偶数，则表示该内核是一个可放心使用的稳定版；如果次版本号是奇数，则表示该内核加入了一些测试的新功能，是一个内部可能存在 BUG 的测试版。如：2.5.74 表示是一个测试版的内核，2.6.12 表示是一个稳定版的内核。

1.3 Linux 桌面环境

Linux 下遵守 X Window 协议有两大类常用的桌面应用环境 GNOME 与 KDE。桌面环境是操作系统与应用软件之间的部分，具体的应用软件都能够在两个桌面环境应用，只是开发环境不同，默认设置有所差异，有时候需要设定环境参数。

1.3.1 KDE 简介

KDE 即 K 桌面环境（Kool Desktop Environment）的缩写。整个系统采用的都是原 TrollTech 公司（后被 Nokia 所收购）所开发的 Qt 程序库。其目的是在 X Window 上建立一个与 MacOS 或者微软的 Windows 类似的完整易用的桌面环境，从而使 UNIX 更接近广大普通用户。KDE 不仅提供了一个方便易用的超级桌面环境，而且还提供了一套免费的计算开发平台。可以访问 <http://www.kde.org> 查询更多的资料。图 1-4 是 KDE 系统的标志。

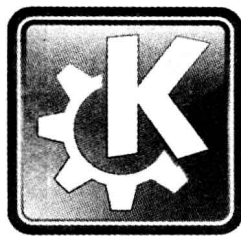


图 1-4 KDE 桌面环境标志

图 1-5 显示了两个 KDE 典型桌面环境界面。

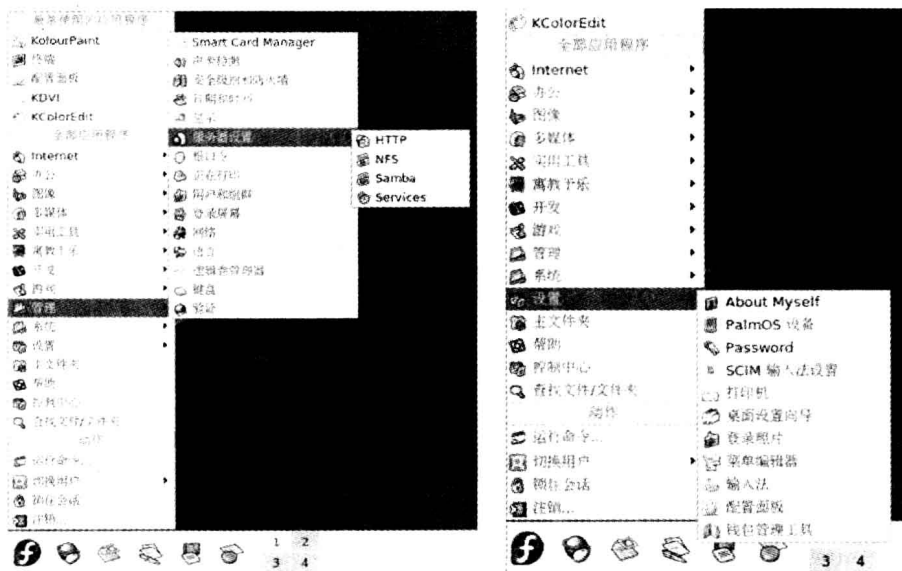


图 1-5 KDE 桌面环境管理

1.3.2 GNOME 简介

GNOME 即 GNU 网络对象模型环境 (The GNU Network Object Model Environment)，是 GNU 计划的一部分，开放源码运动的一个重要组成部分。它是一种让使用者容易操作和设定计算机环境的工具。GNOME 项目有两个目标：提供一个完整的、易学易用的桌面环境——GNOME 桌面环境，为程序设计人员提供强大的应用程序开发环境——GNOME 开发平台，用于开发桌面应用。GNOME 计划是 1997 年 8 月由 Miguel de Icaza 和 Federico Mena 发起，作为 KDE 的替代品。可以访问 <http://www.gnome.org/gnome-3/> 获取更多的资料。图 1-6 是 GNOME 的标志。

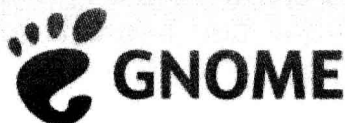


图 1-6 GNOME 标志

图 1-7 是一个典型的 GNOME 桌面环境的界面。

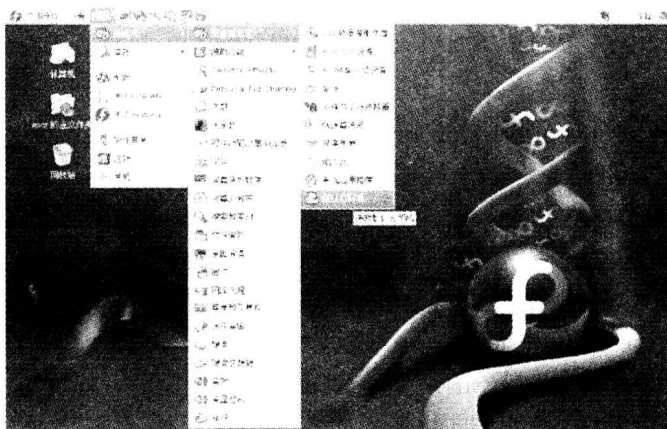


图 1-7 典型的 GNOME 桌面环境界面

GNOME 操作界面由 GNOME 面板 (Panel) 和桌面组成。图 1-8 显示了 GNOME 的面板。

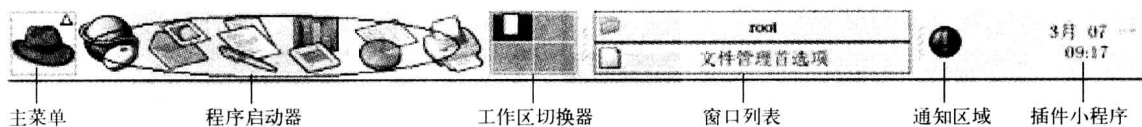
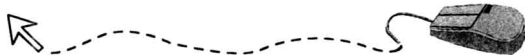


图 1-8 GNOME 面板

GNOME 面板包括以下几部分。

- (1) 主菜单：主菜单是系统中所有应用程序的起点。
- (2) 程序启动器：是 Linux 应用程序的启动链接，如同 Windows 中的快捷方式。
- (3) 工作区切换器：可以将 GNOME 的桌面分为相互独立的工作区，每个工作区是桌面的一部分。
- (4) 窗口列表：窗口列表显示了当前工作区上运行着的应用程序的名称。
- (5) 通知区域：网络更新通知工具是通知区域的一部分。它提供了一种简捷的系统更新



方式，确保系统时刻使用 Red Hat 的最新勘误和错误修正来更新。

(6) 插件小程序 (Applets): 插件小程序是完成特定任务的小程序。GNOME 有很多十分有用并且非常有趣的插件小程序，例如，电子邮件检查器、时钟日历、CPU 和内存情况查看器等。

图 1-9 显示了 GNOME 的文件管理器。文件管理器主要由菜单栏、工具栏、位置栏、状态栏、侧栏和浏览窗口等组成。

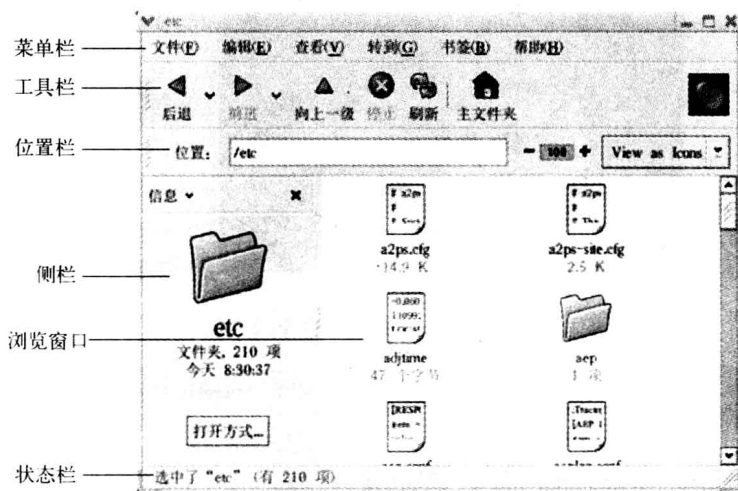


图 1-9 GNOME 文件管理器

1.4 Linux 的发行版本

Linux 主要作为 Linux 发行版（通常被称为“distro”）的一部分而使用。这些发行版由个人、组织松散的团队，以及商业机构和志愿者组织编写。它们通常包括了其他的系统软件和应用软件，一个用来简化系统初始安装的安装工具，以及让软件安装升级的集成管理器。大多数系统还包括了像提供 GUI 界面的 XFree86 之类的曾经运行于 BSD 的程序。

一个典型的 Linux 发行版包括：Linux 内核，一些 GNU 程序库和工具，命令行 shell，图形界面的 X Window 系统和相应的桌面环境，如 KDE 或 GNOME，并包含数千种从办公套件、编译器、文本编辑器到科学工具的应用软件。下面列出了一些常见的 Linux 发行版。

1.4.1 Fedora

Fedora（原名 Fedora Core，自第七版后更名为 Fedora）。它是一套从 Red Hat Linux 发展出来的免费 Linux 系统。Fedora 是一个开放的、创新的、前瞻性的操作系统和平台。它允许任何人自由地使用、修改和重发布，无论现在还是将来。它由一个强大的社群开发，这个社群的成员以自己的不懈努力，提供并维护自由、开放源码的软件和开放的标准。Fedora 项目由 Fedora 基金会管理和控制，得到了 Red Hat, Inc. 的支持。Fedora



图 1-10 Fedora 标志

是一个独立的操作系统，可运行的体系结构包括 X86（即 i386-i686），X86_64 和 PowerPC。图 1-10 是 Fedora 的标志。用户可以访问 <http://fedoraproject.org/> 获取最新的 Fedora 资源。图 1-11 和图 1-12 展示了 Fedora 的最新特性。

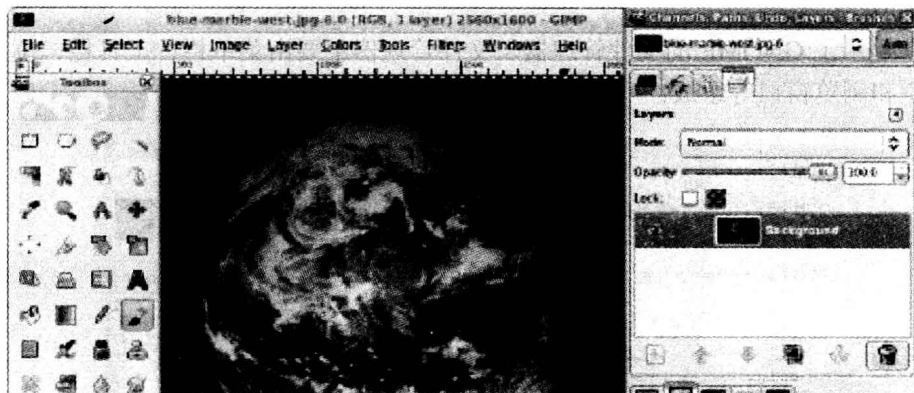


图 1-11 Fedora 图片编辑支持

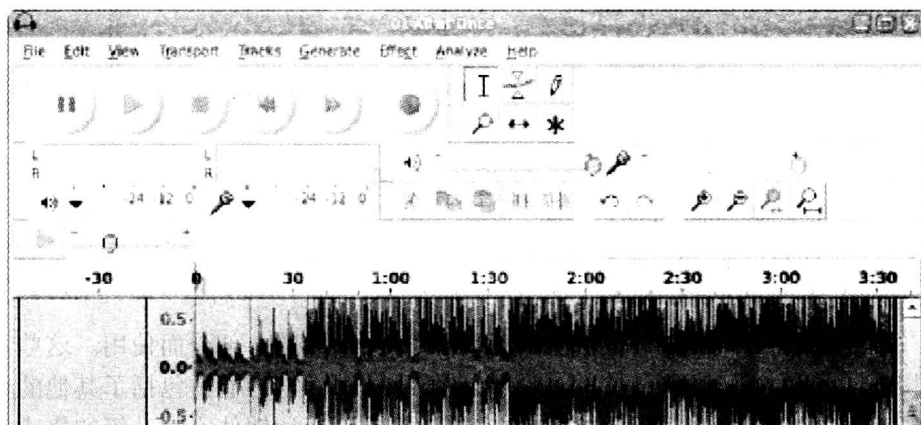


图 1-12 Fedora 音频编辑支持

1.4.2 Debian

Debian Project 诞生于 1993 年 8 月 13 日，它的目标是提供一个稳定容错的 Linux 版本。支持 Debian 的不是某家公司，而是许多在其改进过程中投入了大量时间的开发人员，这种改进吸取了早期 Linux 的经验。Debian 以其稳定性著称，Debian 中大量使用了 pluggable authentication modules(PAM)，综合了一些更易于处理的需要认证的软件（如 winbind for Samba）。Debian 的安装完全是基于文本的。Debian 主要通过基于 Web 的论坛和邮件列表来提供技术支持。作为服务器平台，Debian 提供一个稳定的环境。为了保证它的稳定性，开发者不会在其中随意添加新技术，而是通过多次测试之后才选定合适的技术加入。用户可以访问 <http://www.debian.org/> 获取更多的资料。图 1-13 是 Debian 的标志。而图 1-14 展示了最新版的 Debian 界面。



图 1-13 Debian 标志

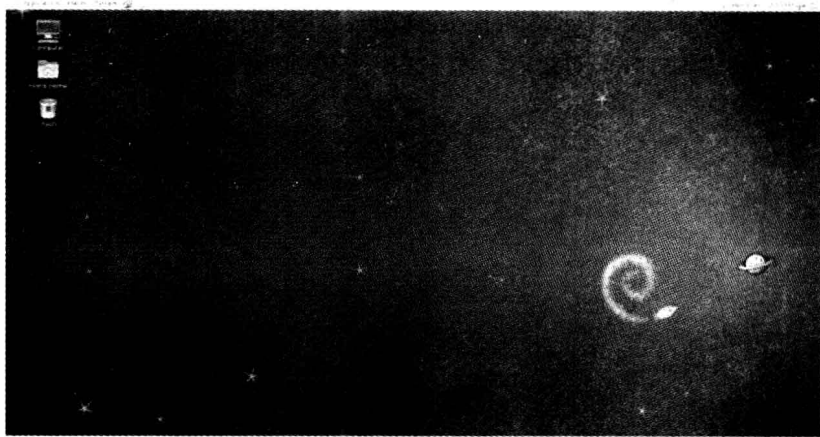


图 1-14 Debian 界面

1.4.3 Ubuntu

Ubuntu 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。图 1-15 是 Ubuntu 的标志。Ubuntu 基于 Debian 发行版和 GNOME 桌面环境，与 Debian 的不同之处在于它每 6 个月会发布一个新版本。Ubuntu 的目标在于为一般用户提供一个最新的、同时又相当稳定的主要由自由软件构建而成的操作系统。Ubuntu 具有庞大的社区力量，用户可以方便地从社区获得帮助。用户可以访问 <http://www.ubuntu.org.cn/> 获取最新的资源。图 1-16 展示了 Ubuntu 11.04 的用户界面。



图 1-15 Ubuntu 的标志

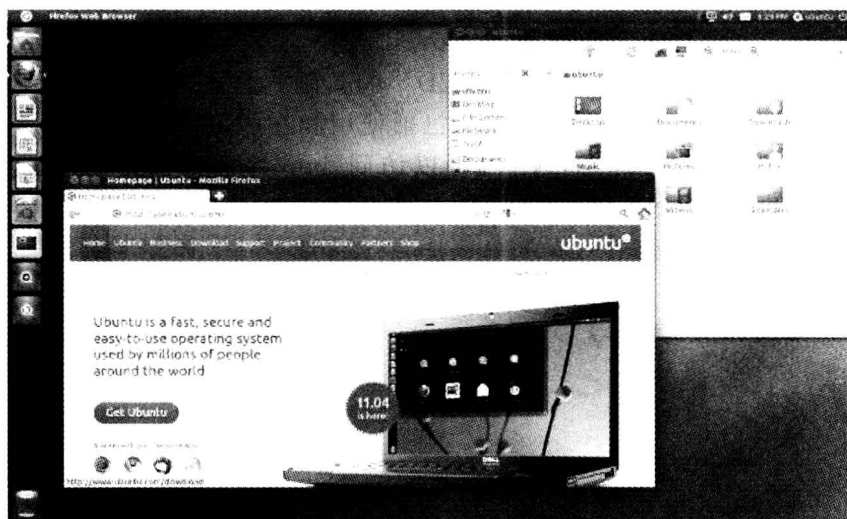


图 1-16 Ubuntu OS 11.04 界面

1.4.4 Red Hat

与其他大部分免费提供的 Linux 发行版本不同，Red Hat 公司主要面向企业用户提供付