



嵌入式开发直通车

Linux C

编程直通车



叶茂 张繁 编著

106个案例+105个上机练习+2个综合项目
+34小时视频教学

超大容量多媒体语音教学视频赠送，
总时长超过34小时



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

[<http://www.phei.com.cn>]



含DVD光盘1张

014032961

TP316.85
45

嵌入式开发直通车

Linux C 编程直通车

叶茂 张繁 编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING



北航

C1721232

TP316.85
45
P

120330310

内 容 简 介

本书以 Fedora 14 为平台,深入浅出、循序渐进地讲解了 Linux 操作系统下的 C 语言程序设计,并通过大量的程序实例,以及综合开发案例的演示,帮助读者快速掌握 Linux 下 C 语言编程的方法和技巧。

本书内容翔实,包括 Linux 系统简介、Shell 编程、Linux C 开发环境、Linux 下常用 C 库函数、进程管理、线程管理、基于文件描述符的 I/O 操作、基于流的 I/O 操作、套接字编程、Linux 数据库编程、GTK+ 图形界面开发等,并结合两个具体的项目案例,详细介绍了 Linux 平台下 C 程序开发的方法和技巧。

本书由作者根据多年来的开发工作经验编著而成,语言通俗易懂,内容丰富,注重实例讲解,知识涵盖面广。本书既适合 Linux 平台下 C 语言编程的初学者及高校本科生、研究生阅读,也适合在 Linux 系统下进行 C 程序开发的工程师查阅和学习。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Linux C 编程直通车 / 叶茂, 张繁编著. —北京: 电子工业出版社, 2014.2

(嵌入式开发直通车)

ISBN 978-7-121-22249-8

I. ①L… II. ①叶… ②张… III. ①Linux 操作系统—程序设计 ②C 语言—程序设计 IV. ①TP316.89 ②TP312

中国版本图书馆 CIP 数据核字 (2013) 第 315148 号

策划编辑: 王敬栋 (wangjd@phei.com.cn)

责任编辑: 徐 萍

印 刷: 涿州市京南印刷厂

装 订: 涿州市京南印刷厂

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: 29 字数: 742 千字

印 次: 2014 年 2 月第 1 次印刷

印 数: 3 000 册 定价: 88.00 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

自 Linux 操作系统诞生以来, 几乎每天, 它都会以某种方式出现在我们的生活中, 我们已经数不清有多少应用程序和有多少机构在使用 Linux。在大中型服务器解决方案中, 国内外无数大型企业都选择了 Linux; 在嵌入式开发领域, Linux 的应用更是层出不穷; 尤其是在当今快速发展的智能手持设备中, 更是以源码开放的 Linux 内核作为平台。基于 Linux 平台的软件开发工程师的需求量也在不断增加。

作者根据自己多年来在 Linux 下进行 C 程序开发的经验与积累, 并融合大量的程序实例而著成此书。本书由浅入深, 适合各个水平阶段的读者学习。

本书特点

1. 循序渐进, 由浅入深

从 Linux 系统的安装、C 语言编程基础、Linux 下的基本编辑器、程序编译器和调试器, 到 Linux 系统的各种函数调用, 再到 Linux 下具体程序案例的设计开发, 本书内容由浅入深, 囊括了 Linux 下 C 程序开发的各个环节。

2. 程序实例丰富, 实践性强

在本书中, 几乎每个知识点都会伴随一个或多个程序实例, 通过实例来加深读者对知识点的理解和掌握。对于每个程序实例, 作者都添加了十分详细的注释, 方便读者理解; 并且, 所有的实例读者都可以在自己的实验环境中完整实现。尤其是通过最后两章实际项目案例的讲解, 更是完整地向读者演示了 Linux 环境下项目实例的设计与开发。

3. 技术全面, 知识点阐述到位

网络编程和图形界面编程是 C 程序学习中比较深入的知识, 也是本书中重点讲解的内容。在本书的最后 3 章, 将这些知识点与实际的项目开发结合, 通过逐步设计与实现, 深化读者对它们的理解与掌握。将 Linux 系统开发技术、C 语言开发技术、软件工程思想融会贯通, 使本书成为一本思想和内容都极其丰富的计算机教程。

本书内容

第 1 章: 介绍 Linux 系统的基本概念和安装方法、Linux 下的常用命令, 以及 Shell 的初步使用。这些是使用 Linux 的基础, 帮助读者为本书后续的学习打下扎实的基础。

第 2 章: 讲述 Linux 下的 Shell 编程方法。Shell 作为 Linux 用户与内核之间交互的接口, 除了是一个命令解释器外, 本身也是一种功能强大的程序设计语言。掌握好 Shell 程序的基本语法, 将会使我们在 Linux 下的程序开发事半功倍。

第 3 章: 介绍 Linux 系统下的 C 开发环境, 包括编辑器、程序编译器和程序调试器, 它们是在 Linux 下进行 C 程序开发必备的工具, 几乎所有 Linux 下的项目编程都会涉及它们。

第 4 章: 详细讲解了 Linux 下的常用 C 库函数。灵活地使用库函数, 将使我们的程序更加高效、简洁。

第 5 章: 详细阐述了 Linux 下的进程管理。进程是操作系统中一个非常重要的概念, 熟悉使用进程的操作和进程控制的相关系统调用, 会使用户在使用 Linux 系统完成各种工作时

更加得心应手。

第 6 章：详细阐述了 Linux 下的线程管理。线程是一个进程内的基本调度单位，是一种轻量级的进程，创建一个线程的开销比创建一个进程要小得多，而且线程上下文的切换开销也比进程要小得多。多线程编程已成为开发并发应用程序的主流方法。

第 7 章：讲解基于文件描述符的文件 I/O 操作，以及 Linux 中文件系统的概念。文件操作是 Linux 系统中最常见的操作之一，在 Linux 中，所有的内容都被看成文件，所有的操作都可以归结为对文件的操作。

第 8 章：讲述基于流的文件 I/O 操作。基于流的 I/O 操作是由标准 C 函数库提供的，与基于文件描述符的 I/O 操作相比，基于流的 I/O 更简单、方便。在大多数情况下，程序员更愿意使用基于流的输入/输出方法。

第 9 章：详细讲述了 Linux 套接字编程的原理与方法，通过大量的程序实例演示了 Socket 编程中常用 API 的使用方法。

第 10 章：详细讲述了 Linux 下 MySQL 数据库编程的原理与方法，通过大量的程序实例演示了 MySQL 数据库编程中常用 API 的使用方法。

第 11 章：讲述 GTK+ 图形界面编程。GTK+ 是 Linux 下基于 C 的图形界面开发库，本章通过程序实例演示了使用 GTK+ 库创建各种界面控件的方法。

第 12 章：通过设计一款 Linux 下的小游戏——俄罗斯方块，演示了小型项目工程软件的模块划分方法，以及 Linux 下的 C 程序开发步骤，使读者能够更深层次地掌握 GTK+ 图形界面编程。

第 13 章：通过设计 Linux 下的学生信息管理系统，再次详细阐述了 Linux 下 C 程序的开发，读者从中能够掌握 GTK+ 图形界面开发和 MySQL 数据库编程的方法与技巧。

本书适合的读者

- Linux 及 C 程序设计初学者；
- 大、中专院校计算机及相关专业的学生；
- 高校计算机及相关专业本科生、研究生；
- Linux 软件开发从业人员；
- 开源软件开发爱好者；
- 社会培训学生。

本书由叶茂、张繁编著。在本书完成之际，感谢我的家人、朋友在我最困难的时候带给我快乐与动力，支持我一直坚持下来，并最终完成这部著作。

参加本书编写的还有王坚宁、李龙、魏勇、张玉兰、高克臻、张秀梅、张云霞、周兴国、李辉、刘峰、朱丽云、徐浩、马建军、许小荣。在此，对以上人员致以诚挚的谢意！

由于时间仓促，加之水平有限，书中的缺点和不足处在所难免，敬请读者批评指正。

编著者

目 录

第 1 章 Linux 初识	1
1.1 了解 Linux	1
1.1.1 什么是 Linux	1
1.1.2 GNU 与 Linux	2
1.1.3 Linux 系统的特点	3
1.1.4 Linux 的内核版本与发行版本	3
1.1.5 本书使用的 Linux	4
1.2 安装 Linux	4
1.2.1 系统安装前的准备工作	5
1.2.2 从光盘安装 Linux	5
1.2.3 从硬盘安装 Linux	22
1.2.4 在虚拟机下安装 Linux	23
1.3 Linux 常用命令	28
1.3.1 Shell 终端	28
1.3.2 与目录相关的命令	29
1.3.3 与文件相关的命令	29
1.3.4 与网络服务相关的命令	31
1.4 本章小结	31
上机题	32
第 2 章 Shell 编程	33
2.1 Shell 简介	33
2.1.1 关于 Shell	33
2.1.2 Shell 的使用	34
2.1.3 常见 Shell 的种类	36
2.2 Shell 基本语法	38
2.2.1 变量	38
2.2.2 通配符	43
2.2.3 引号	44
2.2.4 注释符	46
2.3 选择结构	46
2.3.1 test 命令	46
2.3.2 if 语句	49
2.3.3 if...else 语句	49
2.3.4 if...elif 语句	50

2.3.5 case 语句	52
2.4 循环结构	54
2.4.1 for 语句	54
2.4.2 while 语句	56
2.4.3 until 语句	58
2.4.4 异常退出循环	58
2.5 函数	61
2.5.1 定义函数	61
2.5.2 交互式 Shell 中定义和使用函数	62
2.5.3 Shell 脚本中定义和使用函数	62
2.5.4 将函数定义在单独的文件中	64
2.6 本章小结	65
上机题	65
第 3 章 Linux C 开发环境	67
3.1 C 编辑器	67
3.1.1 vi	67
3.1.2 gedit	72
3.1.3 KWrite	74
3.1.4 Kate	75
3.2 程序编译器 gcc	78
3.2.1 gcc 简介	78
3.2.2 gcc 编译初步	79
3.2.3 警告提示功能	81
3.2.4 优化 gcc	82
3.2.5 连接库	86
3.2.6 调试选项	87
3.3 程序调试器 gdb	88
3.3.1 gdb 简介	88
3.3.2 gdb 调试初步	90
3.3.3 使用断点	92
3.3.4 查看运行时数据	97
3.3.5 查看源程序	103
3.3.6 改变程序的执行	105
3.4 本章小结	109
上机题	110
第 4 章 Linux 下常用 C 库函数	112
4.1 使用库函数的优势	112
4.2 内存操作相关	114
4.2.1 malloc 函数	115

4.2.2	calloc 函数	115
4.2.3	free 函数	117
4.2.4	memset 函数	118
4.2.5	memcpy 函数	123
4.2.6	memmove 函数	125
4.3	数字与字符串间的转换	126
4.3.1	字符串转换为整型	126
4.3.2	字符串转换为实型	129
4.3.3	整型转换为字符串	130
4.3.4	实型转换为字符串	132
4.4	日期与时间	135
4.5	随机函数	138
4.6	本章小结	142
	上机题	143
第 5 章	进程管理	144
5.1	进程概述	144
5.1.1	Linux 进程简介	144
5.1.2	task_struct 结构	146
5.2	进程管理	150
5.2.1	获得进程 ID	151
5.2.2	创建进程	152
5.2.3	进程执行	159
5.2.4	进程消亡	163
5.2.5	进程等待	167
5.2.6	system 函数	173
5.2.7	进程组与会话	175
5.3	进程调度机制	177
5.4	本章小结	180
	上机题	180
第 6 章	线程管理	182
6.1	线程概述	182
6.2	线程的基本操作	183
6.2.1	创建线程	183
6.2.2	线程等待	186
6.2.3	线程终止	187
6.2.4	线程属性	189
6.3	线程同步	193
6.3.1	互斥锁	193
6.3.2	条件变量	198

6.3.3 信号量	203
6.4 本章小结	207
上机题	208
第7章 基于文件描述符的 I/O 操作	209
7.1 Linux 文件系统	209
7.1.1 文件系统结构	209
7.1.2 文件类型	210
7.1.3 文件访问权限	213
7.1.4 数据结构	214
7.2 文件 I/O 操作	216
7.2.1 文件的创建、打开与关闭	217
7.2.2 文件的定位	221
7.2.3 文件的读/写	224
7.3 文件属性操作	228
7.3.1 改变文件访问权限	228
7.3.2 改变文件所有者	229
7.3.3 重命名	230
7.4 文件的其他操作	233
7.4.1 stat()、fstat()和 lstat()函数	233
7.4.2 dup()和 dup2()函数	235
7.4.3 fcntl()函数	236
7.4.4 sync()和 fsync()函数	236
7.5 特殊文件的操作	237
7.5.1 目录文件	237
7.5.2 链接文件	241
7.5.3 管道文件	244
7.5.4 设备文件	246
7.6 本章小结	246
上机题	246
第8章 基于流的 I/O 操作	248
8.1 流与缓存	248
8.1.1 流和 FILE 对象	248
8.1.2 缓存	250
8.1.3 设置缓存	251
8.2 流的打开与关闭	255
8.2.1 流的打开	255
8.2.2 流的关闭	257
8.2.3 注意 fclose	259
8.3 流 I/O 操作	260

8.3.1	字符 I/O	260
8.3.2	行 I/O	264
8.3.3	直接 I/O	267
8.3.4	格式化 I/O	270
8.4	本章小结	273
	上机题	273
第 9 章	套接字编程	274
9.1	套接字编程基础	274
9.1.1	套接字概述	274
9.1.2	套接字数据结构	276
9.1.3	基础函数	279
9.2	TCP 套接字编程	285
9.2.1	TCP 套接字原理	285
9.2.2	TCP 套接字函数	286
9.2.3	TCP 套接字 Client/Server 程序实例	299
9.3	UDP 套接字编程	304
9.3.1	UDP 套接字原理	304
9.3.2	UDP 数据发送与接收	305
9.3.3	UDP 套接字 Client/Server 程序实例	305
9.4	多客户模型	309
9.4.1	循环服务器	309
9.4.2	并发服务器	310
9.4.3	多路复用 I/O	311
9.5	本章小结	319
	上机题	320
第 10 章	Linux 数据库编程	321
10.1	Linux 数据库简介	321
10.1.1	数据库的基本概念	321
10.1.2	关系型数据库	322
10.1.3	Linux 下的常用数据库	324
10.2	MySQL 的使用	326
10.2.1	安装 MySQL	326
10.2.2	MySQL 的常用操作	331
10.3	数据库编程	334
10.3.1	数据结构	334
10.3.2	函数 API	338
10.3.3	一个简易的员工信息管理系统	353
10.4	本章小结	358
	上机题	358

第 11 章	GTK+图形界面开发	359
11.1	Linux 图形界面开发简介	359
11.1.1	Linux 图形界面环境	359
11.1.2	GTK+简介	360
11.2	基本控件	363
11.2.1	窗口	363
11.2.2	标签	368
11.2.3	按钮	369
11.2.4	文本框	371
11.3	布局控件	378
11.3.1	表格	378
11.3.2	框	382
11.3.3	窗格	383
11.4	其他高级控件	385
11.4.1	进度条、微调按钮、组合框	385
11.4.2	单选按钮、复选按钮	390
11.4.3	下拉菜单	392
11.5	GTK+信号与事件	394
11.6	本章小结	398
	上机题	398
第 12 章	设计一款 Linux 下的小游戏	400
12.1	游戏简介	400
12.2	程序模块划分	403
12.2.1	初始化模块	403
12.2.2	对话框显示模块	405
12.2.3	游戏主模块	408
12.3	游戏效果演示	426
12.4	本章小结	427
第 13 章	设计 Linux 下的学生信息管理系统	428
13.1	软件功能简介	428
13.2	程序模块划分	429
13.2.1	数据库设计	429
13.2.2	用户登录	436
13.2.3	学生查询界面	441
13.2.4	教师查询界面	443
13.2.5	管理员查询界面	445
13.2.6	Makefile 文件	448
13.3	软件运行效果演示	448
13.4	本章小结	452

第1章

Linux 初识

Linux 是一种免费的、提供源代码的、适用于 PC 的、类似于 UNIX 的网络操作系统。Linux 来源于 UNIX，它是在 Internet 上由志愿者开发的与 UNIX 兼容的完整的计算机系统，Linux 可以从许多以电子形式发布的提供者那里免费获得。本章将简单介绍 Linux 操作系统的基础知识，以帮助读者对所要学习的知识建立起清晰的认识，为以后的学习打下坚实的基础。

本章内容：

- 了解 Linux；
- Linux 系统的安装；
- Linux 的常用命令。

1.1 了解 Linux

Linux 是一种免费的计算机系统，它起源于 UNIX 操作系统，Linux 的发展与 GNU 密不可分。Linux 是真正的多用户、多任务、多平台的操作系统，它的源代码是开放的，任何人都能修改和重新发布它。现在我们就一起开始 Linux 探索的神秘之旅吧。

1.1.1 什么是 Linux

Linux 是一种计算机系统，Linux 操作系统的内核的名字也是“Linux”。Linux 是当今 UNIX 世界中最激动人心的进展之一。开放源代码的 Linux 是目前可以免费得到的可靠性极高的操作系统之一。Linux 可以从许多以电子形式发布的提供者那里免费获得，它的软件包中包括 X Window 系统、TCP/IP 网络功能、流行的软件工具（如 Emacs 和 Tex）、完整的软件开发环境（包括 C、C++、Perl、Tcl/Tk），还有库、除错器、多媒体的支持，以及科学和数据库应用，等等。

Linux 内核最初是为英特尔 386 微处理器而设计的，现在 Linux 内核支持从 PC 到大型主机甚至包括嵌入式系统在内的各种硬件设备。Linux 系统是由世界各地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版权制约的、全世界都能自由使用的

UNIX 兼容产品。

Linux 操作系统是由 UNIX 发展而来的，UNIX 操作系统是 1969 年由 Ken Thompson 和 Dennis Ritchie 在美国贝尔实验室开发的一种操作系统。由于其良好而稳定的性能迅速在计算机中得到广泛的应用，在随后几十年中也有了不断的改进。

在 20 世纪 80 年代，Andrew S.Tanenbaum 为了满足教学的需要编写了一个与 UNIX 类似的 Minix 系统。1990 年，芬兰人 Linus Torvalds 接触了 Minix 系统后，开始着手研究编写一个开放的与 Minix 系统兼容的操作系统。1991 年 10 月 5 日，Linus Torvalds 在赫尔辛基技术大学的一台 FTP 服务器上发布了一个消息，这也标志着 Linux 系统的诞生。

在开始的时候，Linux 只是个人狂热爱好的一种产物。但是现在，Linux 已经成为一种受到广泛关注和支持的操作系统。包括 IBM 和惠普在内的一些计算机业巨头也开始支持 Linux。很多人认为，和其他的商用 UNIX 系统及微软 Windows 相比，作为自由软件的 Linux 具有低成本、高安全性、更加可信赖的优势。

1.1.2 GNU 与 Linux

我们在阅读有关 Linux 的文章时，经常会看到在 Linux 名称前面有一个前缀“GNU”，类似于“GNU/Linux”。可别小看了这个前缀，如果没有它，或许就没有今天的 Linux。

GNU 计划又称革奴计划，GNU 是“GNU is Not Unix”的缩写，是由自由软件的积极提倡者理查德·斯托曼（Richard Stallman）在 1983 年 9 月 27 日公开发起的，它的目的是希望重现当年软件界合作、开放、团结、互助的精神，而具体的目标则是开发一套区别于当时的商业化 UNIX 系统，但又与之完全兼容的、自由的操作系统。



说明

英文单词“Gnu”原意为非洲牛羚，而有趣的是，GNU 项目的 Logo 采用的就是牛羚的头像。

为保证 GNU 软件可以自由地使用、复制、修改和发布，所有 GNU 软件都有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，该条款称为 GNU 通用公共许可证（GNU General Public License，GPL）。

GPL 的内容主要是保证软件的免费使用和传播，要求必须以源代码的形式发布软件，并且任何使用者都可以以源代码的形式复制或传播软件给任何人。Linus Torvalds 在 1993 年将 Linux 系统转向 GPL，并加入了 GNU。从而最终使自由软件有了发展根基，即基于 Linux 系统的 GNU。这一版权除了规定有自由软件的各项许可权外，还允许用户出售自己的程序拷贝。

绝大多数基于 Linux 内核的操作系统使用了大量的 GNU 软件，包括 shell 程序、工具、程序库、编译器及工具，还有许多其他程序，如 Emacs。正因为如此，GNU 计划的开创者 Richare Stallman 博士提议将 Linux 操作系统改名为“GNU/Linux”，但有些人还是愿意将操作系统称为“Linux”。

换句话说，Linux 只是一个操作系统内核而已，而 GNU 提供了大量的自由软件来丰富在其之上的各种应用程序。

因此,严格来讲,Linux 这个词本身只表示 Linux 内核,但在实际上人们已经习惯了用 Linux 来形容整个基于 Linux 内核,并且使用 GNU 工程的各种工具和数据库的操作系统。基于这些组件的 Linux 软件被称为 Linux 发行版。一般来讲,一个 Linux 发行套件包含大量的软件,例如,软件开发工具、数据库、Web 服务器(如 Apache)、X Window、桌面环境(如 GNOME 和 KDE)、办公套件(如 OpenOffice.org),等等。

1.1.3 Linux 系统的特点

Linux 是一个多任务、多用户、具有复杂内核的操作系统,它充分利用了现行 CPU 的任务切换功能,创造了多任务、多用户环境,允许多个用户同时使用一台计算机系统。同时,多个用户能从相同或不同的终端上用同一个应用程序的副本进行工作,真正实现了多用户的并行操作。

Linux 系统是单内核,这种内核比微内核复杂。在这种内核中,大量的功能是放在内核中直接实现的,而在微内核系统中,许多功能是采用服务进程的形式放在内核外实现的。

Linux 支持现有的常见文件系统,如 Linux Ext2、Ext3、FAT16、FAT32、ISO9660 光盘文件系统和 Windows NT 的 NTFS 文件系统等。它具有严紧的文件及目录结构,文件都是按照作用或者性质来存放的,其目录结构是标准的树状结构。此外,Linux 将设备都当成文件来处理。这样,当要使用某一设备时,只需简单读/写该设备文件就行,极大地方便了设备的使用。

Linux 完全支持 POSIX(可移植性操作系统)规范,可以很容易地将 UNIX 下的应用程序移植到 Linux 下。可移植性使 Linux/UNIX 与其他任何机器进行通信成为可能,而不需要增加通信接口。

Linux 系统具有很强的适应性。Windows 操作系统只能运行在 Intel 处理器上,各厂商的 UNIX 只能运行在各自的处理器上,但是 Linux 系统几乎能运行在所有常见的处理器上。Linux 还支持广泛的外部设备,在 Linux 中可以找到几乎所有的设备驱动程序。

Linux 平台下有大量的应用软件,如电子表格、字处理、数据库、联网工具及游戏等。此外,Linux 使用 RPM 包来包装软件,用 rpm 命令可以很方便地安装、查询、卸载软件。Linux 还支持一系列的开发工具,几乎所有的主流程序设计语言都可以移植到 Linux 上。

Linux 系统为用户提供了强大的管理功能,主要包括存储管理、系统用户和用户组管理、进程管理、文件管理等。

1.1.4 Linux 的内核版本与发行版本

在 Linux 操作系统不断发展壮大的同时,Linux 的内核也在迅速地更新着。Linux 内核的官方版本是由 Linus Torvalds 本人维护的。内核的版本号形式为 major.minor.patchlevel。patchlevel 是对当前内核版本的修订次数。例如,kernel2.0.30 表示对内核版本的第 30 次修订。根据约定,次版本号为偶数时表示该内核为稳定发布版本,对它的修订主要是消除各种错误,为其添加新特性;次版本号为奇数时,则表示其为不稳定的开发版本,开发人员在其中添加了新特性。

在普通用户 PC 上的 Linux 主要是作为 Linux 发行版的一部分而使用的。这些发行版由

个人、松散组织的团队、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，以及一个用来简化系统初始安装的安装工具和软件安装升级的集成管理器。发行版为许多不同的目的而制作，包括对不同计算机结构的支持、对一个具体区域或语言的本地化、实时应用和嵌入式系统等，甚至许多版本故意地只加入免费软件。

目前，超过 300 个发行版被积极开发，最普遍使用的发行版有大约 12 个。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。

主流的 Linux 发行版有 Asianux、B2D Linux、Conectiva Linux、Fedora Core、Fedora Linux、Gentoo Linux、Knoppix Linux、Linux From Scratch、Magic Linux、Mandriva Linux、Red Hat Linux、Slackware Linux、SUSE Linux、Turbo Linux、Debian GNU/Linux、Ubuntu Linux 等。

1.1.5 本书使用的 Linux

本书将在 Fedora 14 系统下为读者讲解 Linux C 编程的知识，书中的实例也全部是在该系统下进行编译和调试的。

Fedora 是一个开放的、创新的、前瞻性的操作系统和平台，基于 Linux。它允许任何人自由地使用、修改和重新发布，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、开放源代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 公司的支持。Fedora 是一个独立的操作系统，是 Linux 的一个发行版，可运行的体系结构包括 x86（即 i386~i686）、x86_64 和 PowerPC 等。

Fedora Linux（第 7 版以前为 Fedora Core）是较具知名度的 Linux 发行包之一，由 Fedora Project 社区开发、Red Hat 公司赞助，目标是创建一套新颖、多功能并且自由（开放源代码）的操作系统。Fedora 基于 Red Hat Linux，在 Red Hat Linux 终止发行后，红帽公司计划以 Fedora 来取代 Red Hat Linux 在个人领域的应用，而另外发行的 Red Hat Enterprise Linux（Red Hat 企业版 Linux，RHEL）则取代 Red Hat Linux 在商业应用的领域。

Fedora 的功能对于用户而言，它是一套功能完备、更新快速的免费操作系统，而对赞助者 Red Hat 公司而言，它是许多新技术的测试平台，被认为可用的技术最终会加入到 Red Hat Enterprise Linux 中。Fedora 大约每 6 个月发布新版本，到本书截稿时，Fedora 最新的版本已经是 Fedora 20 了。Fedora 的发展很快，但依然要选定一个系统来进行讲解，那就是 Fedora 14。

1.2 安装 Linux

学习 Linux，最好的方式莫过于动手与实践，本节将针对 Fedora Linux 14 版本，带领读者在 PC 上实现 Linux 操作系统的各种安装方式。

1.2.1 系统安装前的准备工作

在开始安装之前,最好先对系统有一些了解,可以使用计算机说明书,或者在已经安装了 Windows 系统的计算机上使用“我的电脑”→“属性”菜单,从里面了解一些相关的计算机配置信息。本节向读者讲解在安装 Linux 操作系统时需要注意的几个问题。

1. 操作系统的安装顺序

安装 Linux 前首先要考虑计算机内是否已安装有其他的操作系统,是否要让 Linux 与原有的操作系统并存。若让 Windows 系列操作系统与 Linux 操作系统并存,则务必先安装 Windows 系统,再安装 Linux。因为 Windows 系统的安装程序会更改主引导记录,且无法进行各种作业系统的多重启动。假如硬盘上的分区全部采用 NTFS 文件系统,并且不打算删除其中任何分区上的资料,那就必须建立一个 FAT16 或 FAT32 的分区或再准备一个硬盘,因为 Linux 不支持 NTFS 的写入功能,通过 DOS 格式的分区就能够实现 NTFS 文件系统与 Linux 间的信息传输。

2. 硬件环境

由于设计 Linux 的初衷之一就是用较低的系统配置提供高效率的系统服务,所以安装 Linux 并没有严格的系统配置要求,只要是 Pentium 以上的 CPU、64MB 以上的内存、1GB 左右的硬盘空间,就能安装基本的 Linux 系统并且能运行各种系统服务。但是如果要顺畅地运行 X Window,建议使用 128MB 以上内存。对于初学者而言,建议安装前最好为 Linux 做硬盘规划,空出一个 2GB 左右的磁盘分区安装 Linux 系统。

3. 网络配置

若要连接局域网,则需要先查明 IP 地址、子网掩码、域名服务器、网关等相关设定,了解网卡型号,查看网卡是否被支持,等等。

4. 外设型号

常用外设的型号要了解,如鼠标、键盘类型(PS/2 或 COM)、显卡的型号及各项参数等。

1.2.2 从光盘安装 Linux

Linux 系统最简单、方便的安装方法是从 CD-ROM 安装,用户可以享受最人性化的、类似于 Windows 系列的安装界面。只要将计算机设置成光驱引导,把安装光盘放入光驱,重新引导系统,就会出现安装启动界面,如图 1.1 所示。

Fedora 14 提供图形化的安装界面,在如图 1.1 所示的提示中,安装启动界面上有 5 个选项供用户选择。

- 安装一个新的系统,或者为现有的系统升级更新(Install a new system or upgrade an existing system)。这个选项是默认的,选择此选项,将使用图形安装界面安装系统到计算机。
- 使用基本的视频驱动程序来安装系统(Install system with basic video driver)。为兼容比较老式的显卡,Fedora 14 提供了使用基本图形界面来安装系统的功能。也就是说,在安装时只用到基本的图形功能,这是一般显卡都支持的模式,但可能分辨率

较低，显示效果较差。

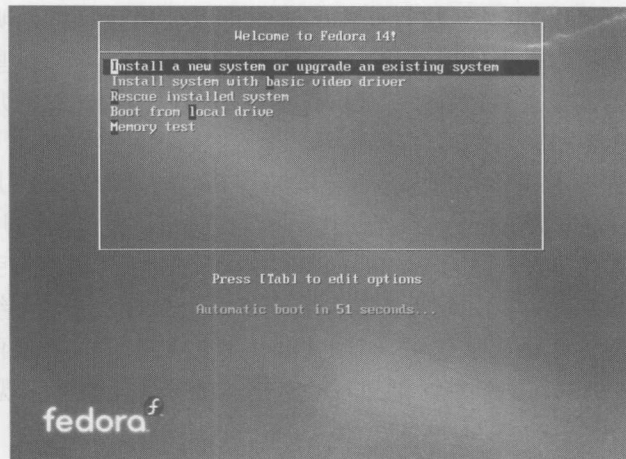


图 1.1 Fedora 14 安装启动界面

- 救援安装的系统 (Rescue installed system)。选择这个选项用来修复安装的 Fedora 14 系统，防止不能正常启动的问题。虽然 Fedora 14 是一个非常稳定的计算机平台，但偶然的问题仍有可能发生。救援环境包含很多实用程序，并会自动为用户修复。
- 从本地驱动器启动 (Boot from local drive)。此选项将退出安装程序并从硬盘启动系统。
- 内存检测 (Memory test)。用来测试当前系统硬件的内存。

这里选择图形化模式安装 Fedora 14，所以直接按回车键即可。

Fedora 14 的安装步骤中比以往多了一个环节，那就是对安装光盘介质的检测，如图 1.2 所示。它允许在开始安装过程前对安装光盘介质进行内容校验，以防止在安装中途由于光盘无法读取或是内容错误造成意外的安装中断，导致前功尽弃。对光盘的检测过程大概会花上十几分钟的时间，也可以单击“Skip”按钮，选择跳过此步。

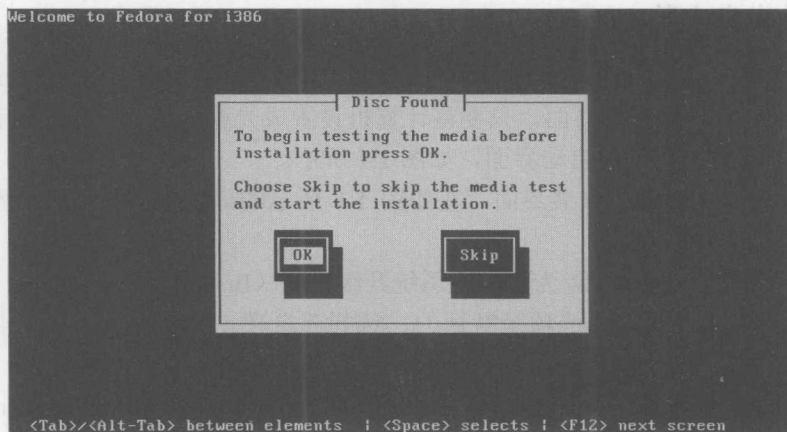


图 1.2 选择 CD-ROM 检测界面