LINUX

- 由浅入深，按理论分析→实际操作→案例分析的顺序组织内容
- 作者多年教学及工程实践的总结，整合了Linux程序设计的绝大多数知识点，涵盖了Linux操作系统下C语言应用程序设计的所有关键内容

高级程序设计

杨宗德 邓玉春 曾庆华 编著

Linux

附源代码光盘

人民邮电出版社
POSTS & TELECOM PRESS

LINUX

高级程序设计

杨宗德 邓玉春 曾庆华 编著

Linux

人民邮电出版社
北京

图书在版编目（CIP）数据

Linux 高级程序设计 / 杨宗德，邓玉春，曾庆华编著. —
北京：人民邮电出版社，2008.1
ISBN 978-7-115-17169-6

Ⅰ. L… Ⅱ. ①杨…②邓…③曾… Ⅲ. Linux 操作系统—
程序设计 Ⅳ. TP316.89

中国版本图书馆 CIP 数据核字（2007）第 175223 号

内 容 提 要

本书以 2.6 内核的 Linux 操作系统为开发平台、GCC 4.0/GDB 6.3 为开发调试环境，详细介绍了 Linux 下 C 语言开发环境、C 语言开发工具、内存管理、ANSI C 文件 I/O 管理、POSIX 文件 I/O 管理、文件及目录管理、进程管理、UNIX 进程间通信机制、System V 进程间通信、多线程编程、线程间通信机制和 Linux Socket 网络编程相关内容及实例开发。

本书内容丰富、紧扣应用，所列代码和实例都来源于具体的应用程序。

本书适合从事 Linux 系统编程工作的人员阅读，也适合从事嵌入式 Linux 开发的人员阅读。

Linux 高级程序设计

◆ 编　　著　杨宗德　邓玉春　曾庆华
责任编辑　刘　浩

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：26.25
字数：623 千字　　2008 年 1 月第 1 版
印数：1–5 000 册　　2008 年 1 月北京第 1 次印刷

ISBN 978-7-115-17169-6/TP

定价：49.00 元（附光盘）

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

在当今流行的通用操作系统阵营中，基本上是 Windows 和类 UNIX 一统江山；在嵌入式领域，Linux 操作系统做为一种专用嵌入式操作系统在嵌入式设备上得到了广泛应用。这主要归功于 Linux 源代码开放、网络功能成熟、极高的安全性、较强的可移植性等特点，另外，在 2.6 内核后，Linux 在实时性上也取得了突破性的发展。这些独特优势得到了人们的青睐，Linux 成为越来越普及的操作系统。

本书所有应用程序采用 2.6 内核为开发平台，GCC 4.0 为开发环境，主要介绍 Linux 操作系统下 C 语言应用程序开发相关内容。

本书主要特点

本书有以下几个特点。

（1）内容丰富。本书是作者多年计算机教学及工程经验总结，整合了 Linux 应用编程的绝大多数知识点，几乎涵盖了 Linux 操作系统下 C 应用编程的所有内容，包括工具使用及环境设置、文件及文件管理、进程及进程管理、进程间通信、线程及线程管理、线程通信、网络及网络应用编程等知识点。

（2）循序渐进。本书在写作思路上避开了大量理论的介绍，按知识体系介绍→应用函数分析→应用案例开发的写作顺序，让读者在掌握具体知识点的同时可以掌握实例的具体实现。

（3）案例指导。本书中所有调用函数及引用都标出具体的出处（在 Linux 操作系统中的文件位置），读者可以一目了然地知道对应函数及类型的定义过程。另外，本书遵循案例教学思想，每一个知识点都讲解一个应用程序，且所有代码都在教学实践过程中调试通过，读者可以直接使用。

（4）紧扣应用。本书所采用的开发平台为 2.6 内核，开发工作为 GCC 4.0，所列代码和实例都来源于具体的应用程序。

本书主要内容

本书共 12 章，详细讲解了 Linux 操作系统下 C 应用程序开发的方方面面。

第 1 章主要介绍 Linux 下 C 语言开发的基本环境。包括 Linux 操作系统编程基本术语介绍、Linux 编程基本概念、Linux 内核及库管理方式以及 Linux 下的编码风格。

第 2 章主要介绍 Linux 下 C 语言开发工具。包括 Linux 开发基本工具的使用、GCC/G++ 编译器、Make 工具及 Makefile 文件、GDB 调试工具以及自动编译调试工具的介绍。

第 3 章主要介绍 Linux 内存管理原理及内存管理工具。包括内存管理的基本概念、Linux 下常用内存管理函数以及常用 Linux 内存管理及调试工具的使用。

第 4 章主要介绍 ANSI C 标准的文件 I/O 管理。包括文件管理基本概念及文件指针、

ANSI C 文件 I/O 管理 API 函数。

第 5 章主要介绍 POSIX 标准的文件 I/O 管理。包括 Linux 系统下文件类型及属性、POSIX 文件 I/O 管理 API 函数。

第 6 章主要介绍 Linux 文件管理及目录操作。包括 Linux 文件系统管理方式、Linux 文件及目录管理操作 API 函数。

第 7 章主要介绍 Linux 进程管理与程序开发。包括进程环境及进程属性、Linux 进程控制、Linux 进程调度等内容。

第 8 章主要介绍 UNIX 进程间通信机制。包括无名管道 PIPE 通信机制、命名管道 FIFO 通信机制和信号中断处理方式。

第 9 章主要介绍 System V 进程间通信。包括 IPC 基础，消息队列、信号量、共享内存等 3 种通信机制的原理及应用。

第 10 章主要介绍 Linux 多线程编程。包括 Linux 线程概述、Linux 线程基本操作、线程属性控制和线程调度等内容。

第 11 章主要介绍线程间通信机制。包括互斥锁、条件变量、读写锁、线程信号等通信机制的原理及应用。

第 12 章主要介绍 Linux Socket 网络编程。包括网络通信基础、Socket 通信基本概念、Socket 通信编程过程，并详细介绍面向链接的 TCP 方式、面向非链接的 UDP 方式、Socket 通信多路选择套接字编程内容。

另外，本书以附录方式介绍了 GCC 参数、GDB 参数、Vim 编辑器参数、Emacs 编辑器参数和 CVS 服务器配置过程等内容。

本书全部代码均在附赠光盘中。

对读者的假定

本书要求读者有较好的 C 语言基础，熟悉 Linux 系统的基本命令。如果读者对操作系统或 Linux 内核有一定了解，则更容易学习本书。

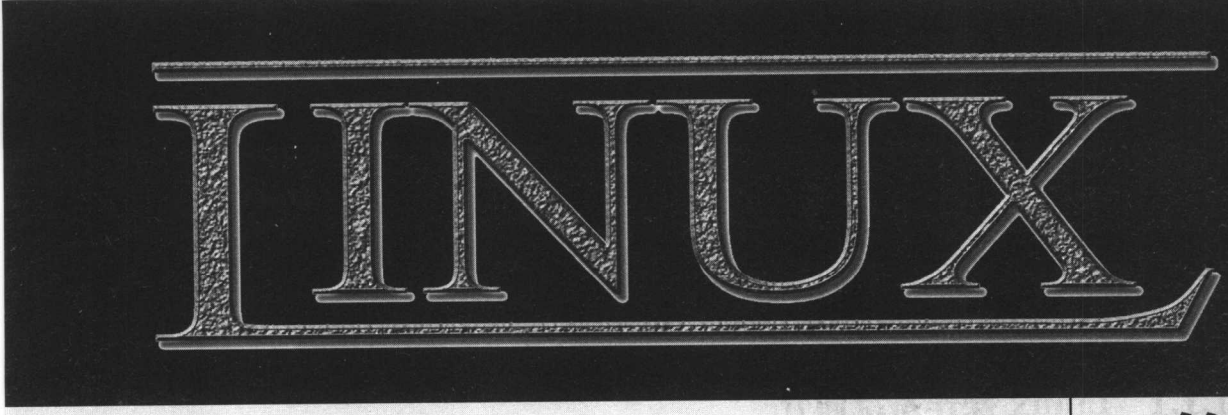
本书编写工作

本书所有内容由杨宗德主编完成，邓玉春、曾庆华参与本书相关代码的编写及审稿工作。同时感谢何伟、张兵、刘兆宏、季建华、刘福刚、赵文革、黄弦等老师对本书的提出了大量宝贵指导意见，另外感谢石昀、朱元斌、钱文杰、陈功杰、汪洪、刘超、钟晓媛、刘梨平、石霞等同学试读了本书初稿，为本书相关内容提出了宝贵意见。由于时间仓促，本书难免有疏忽和不足之处，恳请读者批评赐教（book_better@sina.com）。

编　者

2007 年 11 月

目　录



第1章 Linux下C语言开发环境

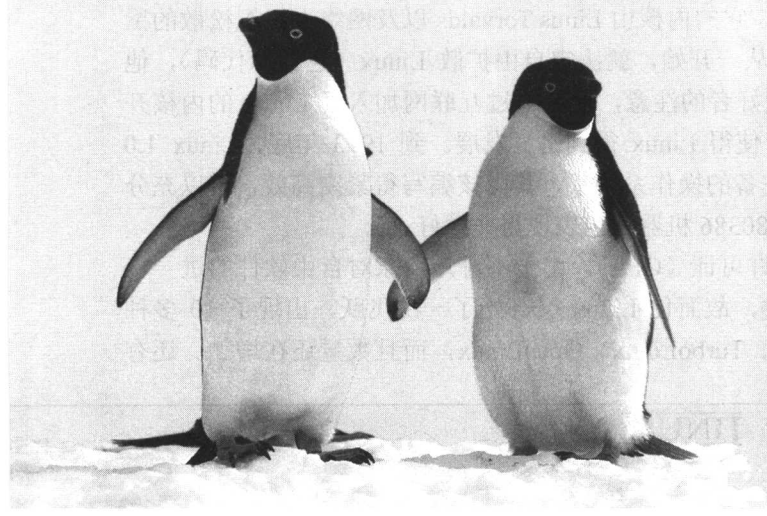
Linux 应用程序开发平台有别于 Windows 应用程序开发平台，因此在介绍具体编程内容之前，本书第 1、2 章介绍 Linux 操作系统下 C 语言的开发环境和开发工具。

本章主要介绍 Linux 下 C 语言开发环境，包括基本概念和基本编程环境。本章第 1 节对 Linux 操作系统及相关术语进行简要介绍。

本章第 2 节主要介绍 Linux 操作系统下编程基本概念以及如何获得 Linux 下的帮助文件。

本章第 3 节主要介绍 Linux 内核目标结构以及库文件管理。Linux 是源码开放的系统，其内核源文件是与发行版本一起发布的，读者可以很方便地查看 Linux 内核处理机制。另外，本节还介绍了 Linux 静态库和共享库的使用和创建方法。

本章第 4 节为读者展示了 GNU 编码规范和 Linux 内核编码规范，遵循这些编码规范不仅可以增强代码的可读性，还能减少代码维护的工作量，提高代码的可扩展性。



1.1 Linux 操作系统概述

1.1.1 Linux 操作系统简介

UNIX 操作系统于 1969 年由 Ken Thompson 在 AT& T 贝尔实验室的一台 DEC PDP-7 计算机上实现。后来 Ken Thompson 和 Dennis Ritchie 使用 C 语言对整个系统进行了再加工和编写，使得 UNIX 能够很容易地移植到以其他硬件为基础的计算机上。由于此时 AT&T 还没有把 UNIX 作为它的正式商品，因此研究人员只是在实验室内部使用并完善它。UNIX 是研究项目，其他科研机构和大学的计算机研究人员也希望能得到这个系统，以便进行自己的研究。AT&T 采用分发许可证的方法，大学和研究机构仅仅需要很少的费用就能获得 UNIX 的源代码。UNIX 的源代码被散发到各个大学，一方面使得科研人员能够根据需要改进系统，或者将其移植到其他的硬件环境中去，另一方面培养了大量懂得 UNIX 使用和编程的学生，这使得 UNIX 的使用更为普及。

到了 20 世纪 70 年代末，在 UNIX 发展到了版本 6 之后，AT&T 认识到了 UNIX 的价值，成立了 UNIX 系统实验室（UNIX System Lab，USL）来继续发展 UNIX。AT&T 一方面继续发展内部使用的 UNIX 版本 7，一方面由 USL 开发对外正式发行的 UNIX 版本，同时 AT&T 也宣布对 UNIX 产品拥有所有权。几乎在同时，加州大学伯克利分校计算机系统研究小组（CSRG）借助 UNIX 对操作系统进行研究，他们对 UNIX 进行的改进相当多，增加了很多当时非常先进的特性，包括更好的内存管理、快速且健壮的文件系统等，大部分原有的源代码都被重写，很多其他 UNIX 使用者，包括其他大学和商业机构，都希望能得到经 CSRG 改进的 UNIX 系统。因此 CSRG 的研究人员把他们的 UNIX 组成一个完整的 UNIX 系统——BSD UNIX（Berkeley Software Distribution）向外发行。

AT&T 的 UNIX 系统实验室，同时也在不断改进其商用 UNIX 版本，直到他们吸收了 BSD UNIX 中已有的各种先进特性，并结合其本身的特点，推出了 UNIX System V 版本。从此以后，BSD UNIX 和 UNIX System V 形成了当今 UNIX 的两大主流，现代的 UNIX 版本大部分都是这两个版本的衍生产品：IBM 公司的 AIX4.0、HP/UX11、SCO 公司的 UNIXWare 等属于 System V，而 Minix、freeBSD、NetBSD、OpenBSD 等属于 BSD UNIX。

Linux 由 UNIX 操作系统发展而来，它的内核由 Linus Torvalds 以及网络上组织松散的黑客队伍一起从零开始编写而成。Linus 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，随即就引起了爱好者的注意，他们通过互联网加入了 Linux 的内核开发工作。一大批高水平程序员的加入，使得 Linux 得到迅猛发展。到 1993 年底，Linux 1.0 终于诞生。Linux 1.0 已经是一个功能完备的操作系统了，其内核编写得紧凑高效，可以充分发挥硬件的性能，在只有 4MB 内存的 80386 机器上也表现得非常好。

Linux 加入 GNU 并遵循公共版权许可证（GPL），由于不排斥商家对自由软件的进一步开发，不排斥在 Linux 上开发商业软件，故而使 Linux 又开始了一次飞跃，出现了 10 多种 Linux 发行版，如 Slackware、Red Hat、TurboLinux、OpenLinux，而且数量还在增加。还有

一些公司在 Linux 上开发商业软件或把其他 UNIX 平台的软件移植到 Linux 上，如今很多 IT 界的大腕如 IBM、Intel、Oracle、Infomix、Sysbase、Netscape、Novell 等公司都宣布支持 Linux! 商家的加盟弥补了纯自由软件的不足和发展障碍，Linux 得以迅速普及。

Linux 操作系统具有以下特点。

- Linux 具备现代一切功能完整的 UNIX 系统所具备的全部特征，其中包括真正的多任务、虚拟内存、共享库、需求装载、优秀的内存管理以及 TCP/IP 网络支持等。
- Linux 的发行遵守 GNU 的通用公共许可证（GPL）。
- 在原代码级上兼容绝大部分的 UNIX 标准（如 IEEE POSIX、System V、BSD），它遵从 POSIX 规范。读者可以在网络上获得关于这一内容的更多信息。

1.1.2 GNU/Linux 简介

GNU（GNU's Not UNIX）工程开始于 1984 年，旨在发展一个类 UNIX 且为自由软件的完整操作系统：GNU 系统。更精确地说，各种使用 Linux 作为内核的 GNU 操作系统都应该被称为 GNU/Linux 系统。

GNU 工程开发了大量用于 UNIX 的自由软件工具和类 UNIX 操作系统，例如 Linux。虽然有许多组织和个人都对 Linux 的发展作出了帮助，但是自由软件基金会依然是最大的单个贡献者。它不仅仅创造了绝大部分在 Linux 中使用的工具，还为 Linux 的存在提供了理论和社会基础。

为保证 GNU 软件可以自由地“使用、复制、修改和发布”，所有 GNU 软件都遵循无条件授权所有权利给任何人的协议条款——GNU 通用公共许可证（GPL，GNU General Public License）。

1985 年 Richard Stallman 又创立了自由软件基金会（FSF，Free Software Foundation）来为 GNU 计划提供技术、法律以及财政支持。尽管 GNU 计划大部分时候是由个人自愿无偿贡献的，但 FSF 有时还是会聘请程序员帮助编写。当 GNU 计划开始逐渐获得成功时，一些商业公司开始介入开发和技术支持。

到了 1990 年，GNU 计划已经开发出的软件包括了一个功能强大的文字编辑器 Emacs、C 语言编译器 GCC 以及大部分 UNIX 系统的程序库和工具。唯一依然没有完成的重要组件就是操作系统的内核（称为 Hurd）。

1.1.3 相关术语介绍

1. POSIX 及其重要地位

POSIX 表示可移植操作系统接口（Portable Operating System Interface，缩写为 POSIX 是为了读音更像 UNIX）。它由电气和电子工程师协会（IEEE，Institute of Electrical and Electronics Engineers）开发，可以提高类 UNIX 环境下应用程序的可移植性。然而，POSIX 并不局限于 UNIX，许多其他的操作系统，例如 DEC OpenVMS 和 Microsoft Windows NT，都支持 POSIX 标准，尤其是 IEEE STD.1003.1-1990（1995 年修订）或 POSIX.1。POSIX.1 给操作系统提供了源代码级别的 C 语言应用编程接口（API），例如读写文件 read/write。POSIX.1 已经被国际标准化组织（ISO，International Standards Organization）所接受，被命名为 ISO/IEC9945-1:1990

标准。POSIX 现在已经发展成为一个非常庞大的标准族，某些部分还处在开发过程中。

2．GNU 和 Linux 的关系

GNU 项目已经开发了许多高质量的编程工具，包括 Emacs 编辑器、著名的 GNU C 和 C++编译器（gcc 和 g++），这些编译器可以在任何计算机系统上运行。所有的 GNU 软件和派生工作均使用 GNU 通用公共许可证，即 GPL。GPL 允许软件作者拥有软件版权，但要授予其他任何人以合法复制、发行和修改软件的权利。

Linux 中使用了许多 GNU 工具，用于实现 POSIX.2 标准的工具几乎都是 GNU 项目开发的。Linux 内核、GNU 工具以及其他的一些自由软件组成了人们常说的 Linux：C 语言编译器和其他开发工具及函数库，X Window 窗口系统，各种应用软件（包括字处理软件、图像处理软件等），其他各种 Internet 软件（包括 FTP 服务器、WWW 服务器）以及关系数据库管理系统等。

3．GPL（General Public License）公共许可协议

GPL 的文本保存在 Linux 系统的不同目录下的 COPYING 文件中。例如，键入“cd /usr/doc/ghostscript*”，然后再键入“more COPYING”可查看 GPL 的内容。GPL 和软件是否免费无关，它的主要目标是保证软件对所有的用户来说都是自由的。GPL 通过如下途径实现这一目标：

（1）要求软件以源代码的形式发布，并规定任何用户都能够以源代码的形式将软件复制或发布给其他用户；

（2）提醒每个用户，对于该软件不提供任何形式的担保；

（3）如果用户的软件使用了受 GPL 保护的软件的任何一部分，该软件都会成为 GPL 软件，也就是说必须随应用程序一起发布源代码；

（4）GPL 并不排斥对自由软件进行商业性质的包装和发行，也不限制在自由软件的基础上打包发行其他非自由软件。

遵照 GPL 的软件并不是可以任意传播的，这些软件通常都有正式的版权，GPL 在发布软件或者复制软件时声明限制条件。但是，从用户的角度考虑，这些根本不能算是限制条件，相反，用户只会从中受益，因为用户可以确保获得源代码。

尽管 Linux 内核也属于 GPL 范畴，但 GPL 并不适用于通过系统调用而使用内核服务的应用程序，通常把这种应用程序看作是内核的正常使用。假如准备以二进制的形式发布应用程序（像大多数商业软件那样），则必须确保自己的程序未使用 GPL 保护的任何软件。如果软件通过库函数调用而且使用了其他软件，则不必受此限制。大多数函数库，受另一种 GNU 公共许可证——LGPL 的保护，下面将会介绍。

4．LGPL（Libraray General Public License）

GNU LGPL（GNU 程序库公共许可证）的内容包括在 COPYINGLIB 文件中。如果安装了内核源程序，在任意一个源程序的目录下都可以找到 COPYINGLIB 文件的一个拷贝。

LGPL 允许在自己的应用程序中使用程序库，即使不公开自己的源代码。但是，LGPL 还规定，用户必须能够获得在应用程序中使用的程序库的源代码，并且允许用户对这些程序库进行修改。

大多数 Linux 程序库，包括 C 程序库（libc.a）都属于 LGPL 范畴。因此，如果在 Linux

环境下，使用 GCC 编译器建立自己的应用程序，程序所连接的多数程序库是受 LGPL 保护的。如果想以二进制的形式发布自己的应用程序，则必须注意遵循 LGPL 有关规定。

遵循 LGPL 的一种方法是，随应用程序一起发布目标代码，以及可以将这些目标程序和受 LGPL 保护的、更新的 Linux 程序库连接起来的 makefile 文件。

遵循 LGPL 另一种方法是使用动态连接。使用动态连接时，即使是程序在运行中调用函数库中的函数时，应用程序本身和函数库也是不同的实体。通过动态连接，用户可以直接使用更新后的函数库，而不用对应用程序重新连接。

1.2 Linux 开发初步

1.2.1 Linux 下 C 程序开发标准

在 Linux 操作系统下进行 C 程序开发的标准主要有两个：ANSI C 标准和 POSIX 标准。

ANSI C 标准是 ANSI（美国国家标准局）于 1989 年制定的 C 语言标准。后来被 ISO（国际标准化组织）接受为标准，因此也称为 ISO C。

POSIX 标准是最初由 IEEE 开发的标准族，部分已经被 ISO 接受为国际标准。

1．ANSI C

ANSI C 的目标是为各种操作系统上的 C 程序提供可移植性保证（例如 Linux 与 Windows 之间），而不仅仅限于类 UNIX 系统。该标准不仅定义了 C 语言的语法和语义，而且还定义了一个标准库。这个库可以根据头文件划分，如表 1-1 所示。

表 1-1　　ISO C 标准定义的头文件

头 文 件	说　明	头 文 件	说　明
<assert.h>	验证程序断言	<signal.h>	信号
<complex.h>	支持复数算术运算	<stdarg.h>	可变参数表
<ctype.h>	字符类型	<stdbool.h>	布尔类型和值
<errno.h>	出错码	<stddef.h>	标准定义
<fenv.h>	浮点环境	<stdint.h>	整型
<float.h>	浮点常量	<stdio.h>	标准 I/O 库
<inttypes.h>	整型格式转换	<stdlib.h>	实用程序库函数
<iso646.h>	替代关系操作符宏	<string.h>	字符串操作
<limits.h>	实现常量	<tgmath.h>	通用类型数学宏
<locale.h>	局部类别	<time.h>	时间和日期
<math.h>	数学常量	<wchar.h>	扩展的多字节和宽字符支持
<setjmp.h>	非局部 goto	<wctype.h>	宽字符分类和映射支持