普通高等教育“十一五”规划教材

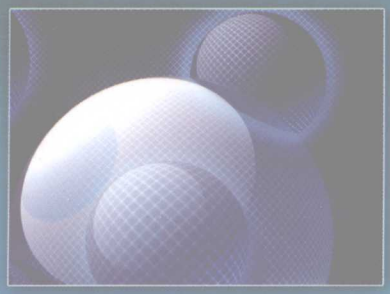
高等院校计算机技术系列教材

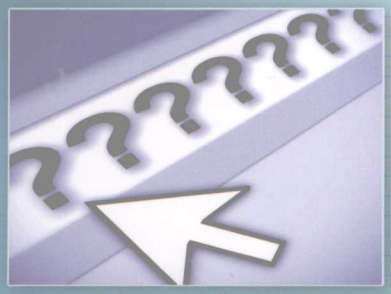
Linux程序员——C语言

刘怀亮 主编

张胜敏 副主编

何国卫 何锦胜 黄建国 苏瑞娟 编著





研究出版社

普通高等教育“十一五”规划教材
高等院校计算机技术系列教材

Linux 程序员——C 语言

刘怀亮　主编

张胜敏　副主编

何国卫　何锦胜　黄建国　苏瑞娟　编著

研究出版社

图书在版编目（CIP）数据

Linux 程序员：C 语言 / 刘怀亮主编.
—北京：研究出版社，2008.4
普通高等教育“十一五”规划教材
高等院校计算机技术系列教材
ISBN 978-7-80168-362-5

I. L…
II. 刘…
III. ①Linux 操作系统—程序设计—高等学校—教材
②C 语言—程序设计—高等学校—教材
IV. TP316.89　TP312

中国版本图书馆 CIP 数据核字（2008）第 049969 号

出版发行　研究出版社
地　址：北京 1746 信箱（100017）
电　话：010-63097512（总编室）010-64045344（发行部）
E-mail：yjcbsfxb@126.com
经　　销　新华书店
印　　刷　广州锦昌印务有限公司
版　　次　2008 年 6 月第 1 版　　2008 年 6 月第 1 次印刷
规　　格　787 毫米 × 1092 毫米　　1/16　　19 印张
字　　数　437 千字
定　　价　43.00 元　　ISBN 978-7-80168-362-5

本书销售专线：010-64045344　64041660

前　言

随着计算机技术的不断发展，计算机技术应用的迅速普及推广，各种软件开发技术也不断地涌现出来。无论是普通的操作系统软件还是为了解决在某个领域中的具体问题而设计的应用软件，都是通过计算机语言编制的程序。在软件设计中，C 语言的使用占了很大的一部分，C 语言是一种结构化、模块化、可编译的通用程序设计语言。它具有表达能力强、可移植性好和代码质量高等特点。C 语言兼有高级语言和低级语言的许多优点，已经成为目前国际上广泛应用的现代主流程序设计语言。本书主要讲述 Linux 系统下 C 语言的应用。

一、关于本书

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

C 语言是一门为计算机软件人员开设的计算机专业基础课程，Linux 作为具有良好特性的操作系统，C 语言作为广泛使用的编程语言，两者的结合为用户提供了广泛的应用前景。掌握 Linux 系统下 C 语言的使用技巧，是软件开发人员最基本的技能。C 语言程序设计是计算机专业的一门重要课程，尤其是在 Linux 系统下的 C 语言应用。

本书是作者依据计算机专业相关教学大纲编写完成的，主要介绍了 Linux 系统下 C 语言的使用和程序设计的方法。

二、本书结构

本书分为 14 章，具体结构安排如下：

第 1 章：Linux 下 C 语言编程简介。主要介绍了 Linux 下 C 语言的发展和特点、C 语言的基础知识、Linux 下 C 语言编程环境。

第 2 章：Linux 程序设计 C 语言基础知识。主要介绍了 C 语言的基础知识、基本数据类型及变量、运算符与表达式、C 语言的语句类型、数据的输入输出设计。

第 3 章：Linux 环境下程序调试基础。主要介绍了 GCC 编译器、GDB 调试器、使用 Make 等相关知识。

第 4 章：选择结构程序设计。主要介绍了 C 语言的选择结构程序设计以及相关实例分析。

第 5 章：循环结构程序设计。主要介绍了 C 语言的循环结构程序设计以及相关实例分析。

第 6 章：数组初步。主要介绍了一维数组、多维数组、字符数组和字符串。

第 7 章：函数的应用。主要介绍了函数的调用、变量的作用范围与存储类别、内部函数和外部函数。

第 8 章：库文件包含及多文件系统的编译。主要介绍了宏定义、库文件包含、头文件和系统帮助。

第 9 章：指针初步。主要介绍了 C 语言所特有的指针与指针变量、指针与函数、指针

与数组的应用。

第 10 章：结构体与共用体。主要介绍了结构体、共用体的定义及使用。

第 11 章：位运算。主要介绍了 C 语言所特有的位运算、位运算符和位段的基础知识及其使用。

第 12 章：文件。主要介绍了文件类型指针、文件的打开与关闭、文件读/写函数、文件的定位以及出错检测。

第 13 章：模拟试题。给出了两套模拟训练题，供读者实战演习、测试和巩固已学知识。

第 14 章：上机实训。给出了十二个很有用的实战训练，每一个实训都包含实训概要、实训内容、实训过程（包括实训分析、实训步骤）及实训总结，供读者上机练习，让读者真正熟悉 Linux 环境下 C 语言程序的编写与调试过程以及 C 语言编程技巧。

三、本书特点

本书知识点全面，深度适宜，语言简练，文笔流畅，结构分明，每章后面给出了练习题，方便读者自学。通过本课程的学习，读者能够熟悉 Linux 操作系统下 C 语言编程环境，掌握 Linux 操作系统下 C 语言编程的基本概念，包括程序编辑环境（vi）、编译工具 GCC、调试工具 GDB、库文件包含及多文件系统的编译、分支程序设计、循环程序设计、一维数组的应用、一维数组与指针、指针数组、标准 I/O 库等。使读者掌握 Linux 操作系统下 C 程序开发的方法和技巧，并具备开发应用程序的能力。

四、本书适用对象

本书涵盖了 Linux 系统基础知识及 C 语言的各个知识点，符合 LUPA 考试认证标准，可作为理工科专业本、专科学生以及职业教育层次学生的理想 Linux C 语言教材，同时也可作为计算机编程爱好者 C 语言程序设计的入门教程。

由于编写时间仓促，加上作者水平有限，书中不足之处和不严谨之处在所难免，恳请专家和读者批评指正。联系方法如下：

电子邮箱：service@cnbook.net　　作者邮箱：great_liu@126.com

网址：www.cnbook.net

本书的电子教案、源代码和习题参考答案可从该网站免费下载，此外，该网站还有一些其他相关书籍的介绍，以供读者选购参考。

编　者

2008 年 3 月

目　录

第 1 章　Linux 下 C 语言编程简介

本章将主要讨论 Linux 下的 C 语言编程环境，包括编辑器、编译器、Make、调试器等，最主要是 VI 编辑器的使用。通过介绍 Linux 下 C 语言编程的基本工具，使编程人员可以很快地进入到 Linux 下编程环境中来。

主要学习内容：

（1）Linux C 简介。

（2）C 语言简介及其特点。

（3）Linux 下 C 语言编程环境。

本章学习目标：

（1）了解 Linux 的发展。

（2）了解 Linux 的特点。

（3）了解 C 语言的特点。

（4）熟悉 Linux 下 C 语言编程环境。

重点和难点：

学习重点：Linux 下 C 语言编程环境和编程风格。

学习难点：Linux 下 C 语言编程环境。

1.1　Linux 的发展和特点

Linux 最初是专门为基于 Intel 处理器的个人计算机而设计的。Linux 的前身是赫尔辛基大学（University of Helsinki）一位名叫 Linus Torvald 的计算机科学系学生的个人项目。Linus 把 Linux 建立在一个基于 PC 机上运行的、缩小型的、名为 Minux 的 UNIX 基础之上，Minux 本身具有 UNIX 的各种特性，这使得以 Minux 做参照而产生的 Linux 继承并更突出了 UNIX 的各种优良特性。当时 Linus Torvold 通过 USENET（新闻组）宣布了 Linux 是一个免费的系统，并指出它主要在 x86 电脑上使用，希望大家一起来将它完善，并将源代码放到了芬兰的 FTP 站点上供人免费下载。本来他想把这个系统称为 freax，可是 FTP 的工作人员认为这是 Linus 的 Minux，就用 Linux 这个子目录来存放，于是它就成了“Linux”。这时的 Linux 只有核心程序（内核），还不能称作是完整的系统，不过由于许多专业用户（主要是程序员）自愿地开发它的应用程序，并借助 Internet 拿出来让大家一起修改一起完善，所以它的周边的程序也越来越多，功能也越来越强大，Linux 本身也就这样逐渐发展壮大起来。

近年来，Linux 操作系统得到了迅猛地发展，在短短的几年之内就包含了 UNIX 的全部功能和特性，在中高端服务器上得到了广泛的应用，国际上很多有名的硬、软件厂商都与之结盟、捆绑，将之用作自己的操作系统。Linux 操作系统得到了非常迅猛地发展，这与 Linux 具有的良好特性是分不开的。

Linux 操作系统的特点可总结为以下几点：

1.　自由软件

Linux 项目从一开始就与 GNU 项目紧密结合起来，它的许多重要组成部分直接来自

GNU 项目。Linux 可以说是作为开放源码的自由软件的代表，便于定制和再开发。在遵从 GPL 版权协议的条件下，各部门、企业、单位或个人就可以免费得到 Linux 源程序，并根据自己的实际需要和使用环境对 Linux 系统进行裁剪、扩充、修改，再开发和发布程序的源码，并公布在 Internet 上。这样就激发了世界范围内热衷于计算机事业的人们的创造力。通过 Internet，这一软件的传播和使用迅速扩大。因为 Linux 操作系统可以从互联网上很方便地免费下载，这样就可以省下购买 Windows 操作系统的一笔不小的资金（正版 Windows 很昂贵）。且由于可以得到 Linux 的源码，所以操作系统的内部逻辑是可见的，这样就可以根据源码准确地查明故障产生的原因，及时采取相应对策。

2. 开放性

开放性是指系统遵循世界标准规范，特别是遵循开放系统互连（OSI）国际标准。凡遵循国际标准所开发的硬件和软件，都能彼此兼容，可方便地实现互连。

3. 多用户

系统资源可以被不同用户各自拥有使用，即每个用户对自己的资源（例如：文件、设备）有特定的权限，互不影响，允许多个用户从相同或不同的终端上同时使用同一台计算机。

4. 多任务

它是指计算机允许多个程序同时执行，而且各个程序的运行互相独立。Linux 系统调度每一个进程，平等地访问微处理器。由于 CPU 的处理速度非常快，其结果是，启动的应用程序看起来好像在并行运行。事实上，从处理器执行一个应用程序中的一组指令到 Linux 调度微处理器再次运行这个程序之间只有很短的时间延迟，用户是感觉不出来的。Linux 充分利用了 X86CPU 的任务切换机制，实现了真正多任务、多用户环境，允许多个用户同时执行不同的程序，并且可以给紧急任务以较高的优先级。

5. 与 UNIX 有良好的兼容性

Linux 是从一个比较成熟的操作系统 UNIX 发展而来的，UNIX 上的绝大多数命令都可以在 Linux 里找到并有所加强。可以认为它是 UNIX 系统的一个变种，因而 UNIX 的优良特点，如可靠性、稳定性以及强大的网络功能，强大的数据库支持能力以及良好的开放性等都在 Linux 上一一体现出来。且在 Linux 的发展过程中，Linux 的用户能大大地从 UNIX 团体贡献中获利，它能直接获得 UNIX 相关的支持和帮助。现在，Linux 已成为具有全部 UNIX 特征、完全符合 POSIX 标准的操作系统。POSIX 是基于 UNIX 的第一个操作系统簇国际标准，该标准最初由 IEEE 开发，部分已经被 ISO 接受为国际标准。POSIX.1 和 POSIX.2 分别定义了 POSIX 兼容操作系统的 C 语言系统接口以及 shell 和工具标准。这两个标准是通常提到的标准。Linux 遵循这一标准使得 UNIX 下许多应用程序可以很容易地移植到 Linux 下，相反也是这样。所有 UNIX 的主要功能都有相应的 Linux 工具和实用程序。对于 UNIX System V 来说，其软件程序源码在 Linux 上重新编译之后就可以运行；而对于 BSD UNIX，它的可执行文件可以直接在 Linux 环境下运行。所以，Linux 实际上就是一个完整的 UNIX 类操作系统。Linux 系统上使用的命令多数都与 UNIX 命令在名称、格式、功能上相同。

6. 性能和稳定性好

在相同的硬件环境下，Linux 可以像其他优秀的操作系统那样运行，提供各种高性能

的服务，可以作为中小型 ISP 或 Web 服务器工作平台。Linux 可以运行在 386 以上及各种 RISC 体系结构机器上。Linux 最早诞生于微机环境，一系列版本都充分利用了 X86CPU 的任务切换能力，使 X86CPU 的效能发挥得淋漓尽致，而这一点连 Windows 都没有做到。Linux 能运行在笔记本电脑、PC、工作站，甚至巨型机上，而且几乎能在所有主要 CPU 芯片搭建的体系结构上运行（包括 Intel/AMD 及 HP-PA、MIPS、PowerPC、UltraSPARC、ALPHA 等 RISC 芯片），其性能远远超过了 Windows NT 操作系统目前所能达到的水平。

7. 可靠的系统安全

Linux 上包含了大量网络管理、网络服务等方面的工具，用户可利用它建立起高效和稳定的防火墙、路由器、工作站、Intranet 服务器及 WWW 服务器。Linux 还包括了大量系统管理软件、网络分析软件、网络安全软件等。由于 Linux 源码是公开的，所以可消除系统中是否有“后门”的疑虑。这对于关键部门、关键应用来说是至关重要的。Linux 采取了许多安全技术措施，包括对读、写进行权限控制、带保护的子系统、审计跟踪、核心授权等，这为网络多用户环境中的用户提供了必要的安全保障。

8. 可移植性好

可移植性是指代码从一种体系结构移植到另外一种体系结构上的方便程度。Linux 是一个可移植性非常好的操作系统，它广泛支持了许多不同体系结构的计算机，能够在从微型计算机到大型计算机的任何环境中和任何平台上运行，包括 Intel、AMD、ARM、Mips 等。可移植性让运行 Linux 的不同计算机平台可以和其他任何机器进行准确而有效的通信，不需要另外增加特殊昂贵的通信接口。

9. 多种用户界面

主要有命令接口、系统调用和图形界面。Linux 的传统用户界面是基于文本的命令行界面，即 shell，它既可以联机使用，又可存在文件上脱机使用。shell 有很强的程序设计能力，用户可方便地用它编制程序，从而为用户扩充系统功能提供了更高级的手段。可编程 Shell 是指将多条命令组合在一起，形成一个 Shell 程序，这个程序可以单独运行，也可以与其他程序同时运行。系统调用给用户提供编程时使用的界面。用户可以在编程时直接使用系统提供的系统调用命令。系统通过这个界面为用户程序提供低级、高效率的服务。Linux 还为用户提供了丰富的图形用户界面，如 GNOME、KDE 等。它利用鼠标、菜单、窗口、滚动条等设施，给用户呈现一个直观、易操作、交互性强的友好的图形化界面。Linux 的 X-Window 可以做 MS Windows 下的所有事情，而且更有趣、更丰富，用户甚至可以在几种不同风格的窗口之间来回切换。

10. 支持多种文件系统

Linux 可以支持十多种文件系统类型：JFS、ReiserFS、ext、ext2、ext3、ISO9660、XFS、Minx、MSDOS、UMSDOS、VFAT、NTFS、HPFS、NFS、SMB、SysV、PROC 等。在 Linux 系统中，每个分区都是一个文件系统，都有自己的目录层次结构，Linux 可以将这些文件系统直接挂载为系统的一个目录。Linux 支持多种文件系统，这样使得它更加灵活，并可以和许多其他种操作系统共存。Virtual File System（虚拟文件系统）使得 Linux 可以支持多个不同的文件系统。由于系统已将 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能同时透明地安装许多不同的文件系统。虚拟文件系统是为 Linux 用户提供快速且高

效的文件访问服务而设计的。随着 Linux 的不断发展，它所支持的文件格式系统也会越来越多。

11. 开发功能强

Linux 支持一系列的 UNIX 开发，Linux 已经具有全部 UNIX 特征，它是一个完整的 UNIX 开发平台，几乎所有的主流程序设计语言都已移植到 Linux 上并可免费得到许多有用的源代码和库，如 C、C++、Fortran77、ADA、PASCAL、Modual2 和 3、Tcl/TkScheme、SmallTalk/X 等。

12. 具有强大的网络功能

Linux 能很好的支持 Internet，免费提供了大量支持 Internet 的软件。Internet 本身是在 UNIX 领域中建立并繁荣起来的，由于 Linux 与 UNIX 的特殊关系，在 Internet 这方面使用 Linux 是相当方便的，用户能用 Linux 与世界上的其他人通过 Internet 网络进行通信。Linux 支持文件传输。用户能通过一些 Linux 命令完成内部信息或文件的传输。Linux 支持远程访问。Linux 不仅允许进行文件和程序的传输，它还为系统管理员和技术人员提供了访问其他系统的窗口。通过这种远程访问的功能，一位技术人员能够有效地为多个系统服务，即使那些系统位于相距很远的地方。实际上，Linux 就是依靠互联网才迅速发展了起来，Linux 具有强大的网络功能也是自然而然的事情。它可以轻松地与 TCP/IP、LAN Manager、Windows for Workgroups、Novell Netware 或 Windows NT 网络集成在一起。Linux 不仅能够作为网络工作站使用，更可以胜任各类服务器，如 X 应用服务器、文件服务器、打印服务器、邮件服务器、新闻服务器等等。

13. 设备独立性

设备独立性是指操作系统把所有外部设备统一当作成文件来看待，只要安装它们的驱动程序，任何用户都可以像使用文件一样，操纵、使用这些设备，而不必知道它们的具体存在形式。具有设备独立性的操作系统，通过把每一个外围设备看作一个独立文件来简化增加新设备的工作。Linux 是具有设备独立性的操作系统，它的内核具有高度适应能力，随着更多的程序员加入 Linux 编程，会有更多硬件设备加入到各种 Linux 内核和发行版本中。

14. 虚拟内存

虚拟内存技术，即拿出一部分硬盘空间来充当内存使用，当内存占用完时，电脑就会自动调用硬盘来充当内存，以缓解内存的紧张。虚拟内存用硬盘空间充当内存来弥补计算机内存空间的缺乏。当实际内存满时（实际上，在内存满之前），虚拟内存就在硬盘上创建了。当物理内存用完后，虚拟内存管理器选择最近没有用过的，低优先级的内存部分写到交换文件上。这个过程对应用是隐藏的，应用把虚拟内存和实际内存看作是一样的，这样就提高了系统的效率。

15. 动态链接共享库

每个应用程序共享一个公用的、运行时可调用的子程序库，而不是保留各自的软件备份，这样可以为系统节省大量空间。在/lib 目录下，就有许多以.so 作后缀的文件，这就是 Linux 系统应用的动态链接库，以 so 结尾，即 Shared Object，共享对象。（在 Linux 下，静态函数库是以.a 作后缀的）X-Window 作为 Linux 下的标准图形窗口界面，它本身就采用了很多的动态链接库（在/usr/X11R6/lib 目录下），以方便程序间的共享，节省占用空间。著名的 Apache

网页服务器，也采用了动态链接库，以便扩充程序功能。这就是动态链接的好处。

1.2 Linux下C语言简介

Linux作为一个优秀的操作系统，一项非常重要的功能就是支持系统调用尤其是支持C语言的系统调用功能十分的方便、快捷。C语言具有高速、灵活、简洁、可移植性好等特点，从而很快成为了世界上最受欢迎的编程语言之一。它和Linux是完美的结合，Linux为C语言提供了很好的支持，为用户提供一个强大的编程环境。事实上，Linux和C语言天生有不解之缘，Linux操作系统是用C语言写的，Linux下的很多软件也是用C语言写的，特别是一些著名的服务软件，比如MySQL、Apache、Oracle等。

1.3 C语言简介及其特点

1963年，剑桥大学将ALGOL 60语言发展成为CPL（Combined Programming Language）语言。1967年，剑桥大学的Matin Richards对CPL语言进行了简化，于是产生了BCPL语言。1970年，AT&T贝尔实验室的Ken Thompson将BCPL进行了修改并设计出更加先进的BCPL并取名为“B语言”。并且他用B语言写了第一个UNIX操作系统。1973年，AT&T贝尔实验室的D.M.Ritchie在B语言的基础上最终设计出了一种新的语言，他取了BCPL的第二个字母作为这种语言的名字，这就是C语言。

随着C语言在各种计算机上的快速推广，从而出现了许多C语言版本。这些版本虽然是类似的，但并不兼容。为了明确定义与机器无关的C语言，1989年美国国家标准协会制定了C语言的标准（ANSI C）。在ANSI标准化后，C语言的标准在相当长的一段时间内都基本保持不变，Normative Amendment1在1995年开发了一个新的C语言版本，但是这个版本很少为人所知。ANSI标准在20世纪90年代又经历了一次比较大的改进，这就是ISO9899:1999（1999年出版）。这个版本就是现在所说的C99，成为现行的C语言标准。

C语言之所以发展迅速，而且成为最受欢迎的语言之一，主要是因为它具有强大的功能。许多著名的系统软件，如UNIX/Linux、Windows、DBASE Ⅲ PLUS、DBASE Ⅳ都是用C语言编写的。用C语言加上一些汇编语言子程序，就更能显示C语言的优势，像PC-DOS、WORDSTAR等就是用这种方法编写的。

总而言之，C语言具有以下特点：

1. C语言是中级语言

C语言被程序员广泛使用的一个重要原因是可以用它代替汇编语言。汇编语言使用的汇编指令，是能够在计算机上直接执行的二进制机器码的符号表示。汇编语言的每个操作都对应为计算机执行的单一指令。虽然汇编语言给予程序员达到最大灵活性和最高效率的潜力，但开发和调试汇编语言程序的困难是难以忍受的。非结构性使得汇编语言程序难于阅读、改进和维护。也许更重要的是，汇编语言程序不能在使用不同CPU的机器间移植。C语言同时具有汇编语言和高级语言的优势，它把高级语言的基本结构和语句与低级语言的实用性结合起来。C语言可以像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元。

2. C语言是结构式语言

结构化语言比非结构化语言更易于程序设计，用结构化语言编写的程序的清晰性使得

它们更易于维护。这已是人们普遍接受的观点了。结构式语言的显著特点是代码及数据的模块化，即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护以及调试。语言是以函数形式提供给用户的，这些函数可方便地调用，并采用多种循环、条件语句控制程序流向，从而使程序完全结构化。在 C 语言中，除实现顺序、选择和循环三种基本结构等的 9 条控制语句外，输入输出操作均由标准库函数（不是 C 语言的组成部分）来实现。所以学习 C 语言，不仅要学习这 9 条控制语句和各种运算符，而且要学习并掌握常用标准库函数的使用。

3. C 语言简洁、灵活，运算符和数据结构类型极其丰富

所有的高级语言都支持数据类型的概念。一个数据类型定义了一个变量的取值范围和可在其上操作的一组运算。常见的数据类型是整型、字符型和实数型。虽然 C 语言有五种基本数据类型，但与 Pascal 或 Ada 相比，它却不是强类型语言。C 程序允许几乎所有的类型转换。例如，字符型和整型数据能够自由地混合在大多数表达式中进行运算。这在强类型高级语言中是不允许的。C 语言的另一个重要特点是它仅有 32 个关键字，这些关键字就是构成 C 语言的命令。和 IBM PC 的 BASIC 相比，后者包含的关键字达 159 个之多。

4. C 语言可移植性好

C 语言程序非常容易移植，可移植性表示为某种计算机写的软件可以用到另一种机器上去。举例来说，如果为苹果机写的一个程序能够方便地改为可以在 IBM PC 上运行的程序，则称为是可移植的。几乎所有的计算机上都有 C 语言编译程序，因此可以很少改动甚至不加改动地将为一种机器写的 C 语言源程序在另一种机器上编译执行。可移植性节省了时间和财力。

5. C 语言生成的目标代码质量高，程序执行效率高

C 语言具有各种各样的数据类型，并引入了指针概念，可以直接操纵硬件。可使程序执行效率更高，但这也使得初学者难于掌握它。用 C 语言编程，可以获得高效机器代码，其效率几乎接近汇编语言代码。

6. C 语言适用范围大

C 语言最初被用于系统程序设计。一个系统程序是一大类程序的一部分，这一大类程序构成了计算机操作系统及实用程序。通常被称为系统程序的有：操作系统翻译程序、编辑程序、汇编程序、编译程序、数据库管理程序。随着 C 语言的普及，加之其可移植性和高效率，许多程序员用它设计各类程序。而且计算功能、逻辑判断功能也强大，可以实现决策目的。C 语言也有强大的图形处理功能，支持多种显示器和驱动器。

1.4 Linux 程序设计基础知识

1.4.1 头文件

glibc_header 是 Linux 下的系统头文件。缺少了系统头文件的话，很多用到系统功能的 C 程序将无法编译。假如用户在安装过程中少装了这些包，就会无法编译 C 源程序。在使用 C 语言和其他语言进行程序设计的时候，需要头文件来提供对常数的定义和对系统及库函数调用的声明。对 C 语言来说，这些头文件几乎永远保存在/usr/include 及其下级子目录里。那些依赖于所运行的 UNIX 或 Linux 操作系统特定版本的头文件一般可以在

/usr/include/sys 或/usr/include/linux 子目录里找到。其他的程序设计软件也可以有一些预先定义好的声明文件，它们的保存位置可以被相应的编译器自动查找到。比如，X 窗口系统的/usr/include/X1R6 子目录和 GNU C++编译器的/usr/include/g++ -2 子目录等。

在调用 C 语言编译器的时候，可以通过给出“ -I”编译命令标志来引用保存在下级子目录或者非标准位置的头文件。用 grep 命令来查找含有某些特定定义与函数声明的头文件是很方便的。

头文件的保存位置：

（1）/usr/include：系统头文件。

（2）/usr/local/include：本地头文件。

1.4.2 函数库

函数库是一些预先编译好的函数的集合，那些函数都是按照可再使用的原则编写的。它们通常由一组互相关联的用来完成某项常见工作的函数构成。比如用来处理屏幕显示情况的函数（curses 库）等。

标准的系统库文件一般保存在/lib 或者/usr/lib 子目录里。编译时要告诉 C 语言编译器（更确切地说是链接程序）应去查找哪些库文件。默认情况下，它只会查找 C 语言的标准库文件。库文件必须遵守一定的命名规则，还必须在命令行上明确地给出来。

库文件的名字永远以 lib 这几个字母打头，随后是说明函数库情况的部分（比如用 c 表示这是一个 C 语言库；而 m 表示这是一个数学运算库等）。文件名的最后部分以一个句点（.）开始，然后给出这个库文件的类型，如下所示：

（1）.a 传统的静态型函数库。

（2）.so 和.sa 共享型函数库（见下面的解释）。

函数库一般分为静态和共享两种格式，用 ls /usr/lib 命令查一下就能看到。在通知编译器查找某个库文件的时候，既可以给出其完整的路径名，也可以使用-l 标志。

1. 静态库

函数库最简单的形式就是一组处于可以“拿来就用”状态下的二进制目标代码文件。当有程序需要用到函数库中的某个函数时，就会通过 include 语句引用对此函数做出声明的头文件。编译器和链接程序负责把程序代码和库函数结合在一起成为一个独立的可执行程序。如果使用的不是标准的 C 语言运行库而是某个扩展库，就必须用-l 选项指定它。

静态库也叫做档案（archive），它们的文件名按惯例都以.a 结尾。比如 C 语言标准库为/usr/lib/libc.a、X11 库为/usr/X11R6/lib/libX11.a 等。

自己建立和维护静态库的工作也并不困难，用 ar（“建立档案”的意思）程序就可以做到，另外要注意的是，应该用 gcc -c 命令对函数分别进行编译。应该尽量把函数分别保存到不同的源代码文件里去。如果函数需要存取普通数据，可以把它们放到同一个源代码文件里并使用在其中声明为 static 类型的变量。

GNU 的 C 函数库，即 glibc，是 Linux 上最重要的函数库，它定义了 ISO C 标准指定的所有的库函数，以及由 POSIX 或其他 UNIX 操作系统变种指定的附加特色，还包括有与 GNU 系统相关的扩展。glibc 基于如下标准：

（1）ISO C: C 编程语言的国际标准，即 ANSI C。

（2）POSIX：GNU C 函数库实现了 ISO/IEC 9945-1:1996（POSIX 系统应用程序编程接口，即 POSIX.1）指定的所有函数。该标准是对 ISO C 的扩展，包括文件系统接口原语、设备相关的终端控制函数以及进程控制函数。同时，GUN C 函数库还支持部分由 ISO/IEC 9945-2:1993（POSIX Shell 和工具标准，即 POSIX.2）指定的函数，其中包括用于处理正则表达式和模式匹配的函数。

（3）Berkeley UNIX：BSD 和 SunOS。GNU C 函数库定义了某些 UNIX 版本中尚未标准化的函数，尤其是 4.2BSD、4.3 BSD、4.4BSD UNIX 系统（即“Berkeley Unix”）以及“SunOS”（流行的 4.2BSD 变种，其中包含有某些 Unix System V 的功能）。BSD 函数包括符号链接、select 函数、BSD 信号处理函数以及套接字等等。

（4）SVID：System V 的接口描述。GNU C 函数库定义了大多数由 SVID 指定而未被 ISO C 和 POSIX 标准指定的函数。来自 System V 的支持函数包括进程间通信和共享内存、hsearch 和 drand48 函数族、fmtmsg 以及一些数学函数。

（5）XPG：X/Open 可移植性指南。GNU C 函数库遵循 X/Open 可移植性指南（Issue 4.2）以及所有的 XSI（X/Open 系统接口）兼容系统的扩展，同时也遵循所有的 X/Open UNIX 扩展。

除 glibc 之外，流行的 Linux 发行版中还包含有一些其他的函数库，这些函数库具有重要地位，例如：

（1）GNU Libtool：GNU Libtool 实际是一个脚本生成工具，它可以为软件包开发者提供一般性的共享库支持。以前，如果源代码包的开发者要利用共享库的优点，则必须为每个软件包可支持的平台编写定制的支持代码。并且还需要设计配置接口，以便软件包的安装程序能够正确选择要建立的库类型。利用 GNU Libtool，则可以简化开发者的这一工作。它在一个单独的脚本中同时封装了与平台相关的依赖性以及用户界面。GNU Libtool 可使每个宿主类型的完整功能可通过一般性的接口获得，同时为程序员隐藏了宿主的特殊性。GNU Libtool 一致性接口是可靠的，用户不必阅读那些晦涩的文档，以便在每个平台上建立共享库。他们只需运行软件包的配置脚本，而由 libtool 完成繁复的工作。

（2）CrackLib：CrackLib 为用户提供了一个 C 语言函数接口，利用这一函数，可避免用户选择容易破解的密码。该函数库可在类似 passwd 的程序中使用。

（3）LibGTop：LibGTop 是一个能够获取进程信息以及系统运行信息的函数库，这些信息包括：系统的一般信息、SYS V IPC 限制、进程列表、进程信息、进程映射、文件系统使用信息等。

（4）图形文件操作函数库：包括 libungif、libtiff、libpng、Imlib、libjpeg 等，可分别用来操作 GIF、TIFF、PNG、JPEG 以及其他一些格式的图形文件。

2. 共享库

静态库的缺点是，如果在同一时间运行多个程序而它们又都使用着来自同一个函数库里的函数时，内存里就会有许多份同一函数的备份，在程序文件本身也有许多份同样的备份。这会消耗大量宝贵的内存和硬盘空间。

许多 UNIX 系统支持共享库，它同时克服了在这两方面的无谓消耗。对共享库和它们在不同系统上实现方法的详细讨论超出了本书的范围，本书把注意力集中在眼前 Linux 环境下的实现方法上。