Linux C 语言实务

- 详尽的 C 语言介绍
- Linux 特有的系统调用函数
- 好用的 Linux 程序开发工具
- 丰富的范例讲解
- GCC 常用参数列表
- 使用 gdb 调试工具

施威铭研究室 著

机械工业出版社
China Machine Press

Linux C语言实务

施威铭研究室　著

机 械 工 业 出 版 社

本书从基础的C语言知识入手，重点介绍Linux下的C语言编程，以及在Linux中编写程序应注意的事项及概念。本书分为2篇，第1篇是C语言入门，带领读者从无到有，进入C语言的世界。第2篇是C语言详解，对C语言的构成方面进行详细的讨论；读者可以详尽了解C语言的语法、Linux特有的系统调用函数、Linux的程序开发工具、GCC常用参数列表以及gdb调试工具的使用，并结合丰富的范例讲解进行实际操作。

本书通俗易懂，理论与实践紧密结合，可以作为初学者的C语言教材，同时也是相关人员学习和使用Linux的参考用书。

图书在版编目（CIP）数据

Linux C语言实务/施威铭研究室著.
—北京：机械工业出版社，2002.8
ISBN 7-111-10773-X

Ⅰ L　Ⅱ 施..　Ⅲ.C语言-程序设计　Ⅳ TP312

中国版本图书CIP数据核字（2002）第061217号

机械工业出版社（北京市百万庄大街22号　邮政编码 100037）
责任编辑：于希莹

北京忠信诚胶印厂印刷·新华书店北京发行所发行
2002年9月第1版·第1次印刷
787mm×1092mm　1/16　·21.75印张·526千字
0001-4000册
定价：29 00元

凡购本图书，如有缺页、倒页、脱页，由本社发行部调换
本社购书热线电话：（010）68993821、68326677-2527
封面无防伪标均为盗版

前　言

近年来虽然不断地涌现出许多新的程序语言，但 C 语言仍然占有举足轻重的地位。C 语言的关键字（keyword）大约 30 个左右，但却可以写出功能庞大的程序；而且其运算式也很简洁，由此写出来的程序十分有效率。C 语言也因此成为许多人初次学习程序语言的第一选择。

Linux 是目前主流的计算机操作系统，它保持着自由与开放的精神，使用者只需用很少的钱就能获得软件。当然，若只想在 Linux 中练习 C 语言，也可不需任何金钱便可获得开发工具。

同时 C 语言也是 Linux 的母语，其操作系统本身主要是由 C 语言编写而成的。Linux 的源程序代码大多是开放的，如果想要了解某个程序到底是如何写成的，大多可以顺利取得程序代码进行分析研究。也正是由于这种开放精神，使得 Linux 成为学习 C 语言的最佳平台。

本书针对 C 语言的初学者，从最基本的 C 语言入门、编译器操作，到整个 C 语言的详解都一一作了说明。虽然是以 Linux 操作系统为平台，但书中仍会针对在 Linux 开发程序中应有的概念进行介绍，因此即使对 Linux 没有深入的了解，也能轻松学会。只要循序渐进地学习本书，必定可以顺利地掌握 C 语言的精髓，进而发挥它强大的功能。

施威铭研究室

目　　录

第 1 篇　C 语言入门

第 2 篇 C 语言详解

第 1 篇　C 语言入门

本篇 3 章的内容，将带领读者进入 C 语言的世界。第 1 章将介绍在 Linux 中开发程序的基本概念，并试着编辑、编译、产生第 1 个程序；第 2 章将对 C 语言做一个入门性的介绍，这才是真正接触 C 程序的一章；第 3 章介绍如何使用函数库及整体开发环境的支持。

第 1 章：本章主要说明开发 C 程序的基本概念。首先说明什么是编译语言及 GCC 编译器，并介绍开发程序所需经过的步骤，使读者对程序的产生流程有个基本的认识；接着实际编写一个 C 程序，并产生可执行文件；最后介绍几个在 Linux 中编写程序的技巧。

第 2 章：本章将介绍 C 程序的基本格式、语句的定义、类型的声明等基本知识，另外对于一些重要的 C 语言概念如函数、指针，也都做了初步的介绍，为的是使读者早日熟悉并巩固这些概念。读完本章后读者应该对 C 程序有进一步的了解，并且已经能够编写简单的 C 程序了。

第 3 章：本章讲解 C 语言的 I/O 函数库，包含标准 I/O 的概念、I/O 的转向，以及字符、字符串函数的使用方法。使用 I/O 的转向，使较短的 C 程序也能发挥很强大的功能。另外，本章还介绍 C 语言的操作环境支持，包括 Preprocessor 指令及 make 的使用方法。

原书空白页

第 1 章

Linux C 语言的开发环境

1.1 编译语言的基本认识

计算机是以一种与人类完全不同的形式工作的，亦即其内部的语言与人类的语言完全不同。如果要驱使计算机工作，就必须以计算机认识的语言（即计算机语言）来下达命令，否则会造成“对牛弹琴”的窘境。

然而，要人类说计算机的语言是很不方便的。为了把人的语言编译成计算机的语言，计算机科学家设计了所谓的计算机程序语言（Computer Programming Language），其主要的方式就是把人类的语言在词汇、文法上均加以约束简化，使编译的工作变得简单，以达到实用、迅速的目的。这就是为什么计算机的程序语言总是有种似曾相识（如 if、for、while... 等）的感觉，但其语法（文法）又显得奇怪（多半限制重重）的原因了。

计算机语言的种类很多，有些语言较接近人类的日常用语，称为高级语言，这种语言较易学习使用。有些语言形式上较接近计算机内部语言，称为低级语言，这种语言不易学习，但效率较高。

1.1.1 编译器与解释器

常见的高级语言可分成两种，一种叫编译语言，一种叫解释性语言。无论是编译或解释性的语言，都需要一个编译程序把人类所下达的命令编译成计算机内部的语言。差异在于编译方式的不同。编译语言的编译程序叫编译器（Compiler），解释性语言的编译程序叫解释器（Interpreter）。

编译器是以整批作业的方式进行编译工作。首先必须完成源程序编写，并以编辑程序输入存为文件后，才能由编译器读入、编译，并且把编译的结果存为文件。这种经编译器编译成的文件，称为目标文件（Object File）。必须在整个编译工作完成，并产生可执行的（计算机看得懂的）目标文件后，才能把此目标文件交给计算机执行（目前大多数的系统还要经过链接处理才能执行）。如图 1-1 所示。常见的编译语言有 C/C++、Delphi、Kylix、Pascal 等。

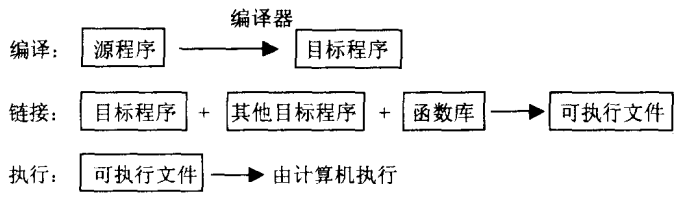

图 1-1 编译语言的处理过程

编译语言的另一项优点是，编译过的程序可以多次使用。通常，都将一些常用的程序

片段写成独立的函数（即子程序），并把这些模块集结成一个文件，在使用时就从此文件中抽取需要的函数模块来链接。这种由常用的函数模块所组成的文件称为函数库（Function Library）。函数库是编译语言的重要概念，将会在后文中详述。

解释器的工作方式与编译器不同。解释器不是采用事前整批作业的方式，而是在程序执行时，以实时的方式进行编译。也就是说，在执行时才把源程序一行一行地读进来进行编译，每读一行编译一行，并送给计算机执行，然后再继续读入下一行。如图 1-2 所示。

解释器与编译器的最大区别是，在每行命令读入、编译、送给计算机执行的过程中，解释器随时控制着整个进程。解释器这种兼具监督执行状况的功能，是编译器所没有的。常见的解释性语言有 Shell script、Python 等。

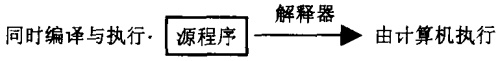

图 1-2 解释性语言的处理过程

1.1.2 解释性语言与编译语言的比较

由于解释性语言不做事先的编译，因此没有编译（及链接）的过程，操作过程简便、迅速。而且由于它具有监督执行状况的功能，所以可以在程序执行的过程中随时指出程序的错误。因此它通常容易学习，且十分友好（Friendly），所以如 Shell script 等这样的解释性语言，特别适合作为初学者的学习语言。

编译语言的优点是事先整批编译，而在执行过程中不必再次编译，所以执行速度很快，通常是解释性语言的几倍到十几倍，特别适于讲求实效性的应用。编译语言的主要缺点是操作过程过于复杂，从程序的编辑、编译、链接到执行，再加上来回调试，无论是方法或时间上都很费力费时。不过近年来出现的集成开发环境（Intergrated Developement Enviroment，IDE）已弥补了这方面的缺点，它将编辑和编译的功能整合在一起，使编译语言的开发过程，和解释性语言一样方便。

目前在 Linux 上最广泛使用的 C 语言编译器是 GCC，而这也是本书中所使用的编译器。

1.2 GCC 简 介

GCC（GNU Compiler Collection）是一套由自由软件基金会（FSF，Free Software Foundation）开发的 C 语言编译器，问世于 1987 年 3 月 22 日，版本编号为具有测试意义的 0.9 版，紧接着在第二天（23 日）立即推出了正式的 1.0 版。经过版本的不断更新，目前最新版本为 3.x 版。其标志如图 1-3 所示。

图 1-3 GCC 的标志

在 Linux 中，GCC 被广泛采用并不是没有理由的，因为它具有以下几个特点：

- 软件获取方便：GCC 已包括在各 Linux 发行版的光盘中。只要在安装操作系统时，同时选择 GCC，那么在装好 Linux 系统后就可以马上使用了。
- 良好的执行效果：GCC 同时整合了编译器与链接器，因此可以将编译与链接同时执行，而不需要再分成 2 个步骤。同时 GCC 产生的可执行文件，在文件大小与执行效率上也颇为理想。
- 具有众多的参数：GCC 执行时可选择的参数很多，借助这些丰富的参数，使编译很有弹性。虽然 GCC 的参数众多，但这并不表示难以使用，一般的程序代码如果不加任何参数，也可顺利编译成功。
- 跨多种操作平台：凭借着 GNU 自由与开放的精神，除了 Linux 外，GCC 还被移植到大多数热门的操作平台上，例如 Solaris、BSD、OS/2，甚至连 Windows、DOS 也可以见到它的踪迹。
- 良好的兼容性：GCC 除了符合标准的 ANSI C 规范之外，还与早期 C 编译器的规范兼容，因此具有绝佳的兼容性。

注意

如果想进一步了解移植到 Windows/DOS 平台的 GCC 相关信息，请参考 cygwin（www.cygwin.com）、MinGW（www.mingw.org）及 Dev-C++（www.bloodshed.net/devcpp.html）等网站的介绍。

1.3 C 程序的开发过程

对 GCC 编译器有了基本的概念之后，接着便可以编写程序了。一个程序从最初的构思到最后完成的过程称为“开发过程”。C 语言程序的开发过程如下：