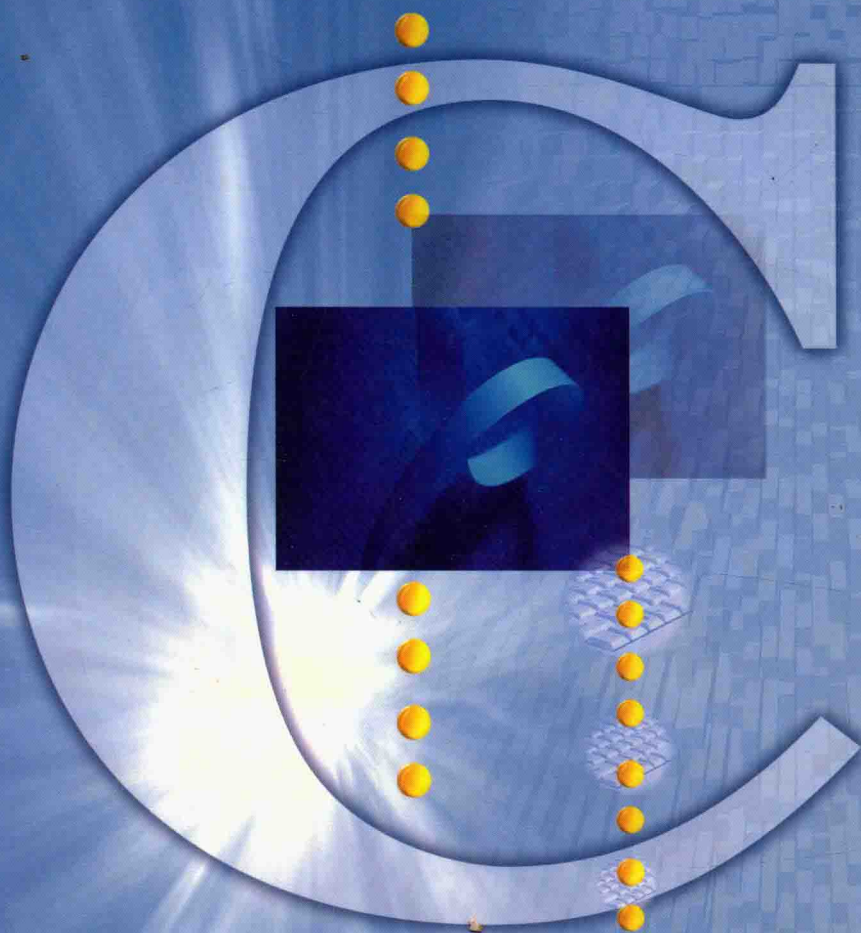


21世纪大学计算机基础课程教材

语言程序设计教程

张毅坤 曹 锰 张亚玲 等 编著
冯博琴 主审



西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

21世纪大学计算机基

基 础 课 内

92/P6

语言程序设计教程

张毅坤 曹 锰 张亚玲 等 编著
冯博琴 主审

西安交通大学出版社
· 西 安 ·

内 容 提 要

本书分为三大部分。第1部分为基础篇,共有8个章节,分别讲述了C语言的基本概念、基本规则与基本内容;第2部分为综合扩展篇,分为4个章节,讲述第1部分未展开的内容,增加了用C语言实现图形和动画的相关基本知识,并从软件工程的角度出发,给出了如何分析问题,解决问题,综合运用C语言实现相对规模较大的两个工程程序的实例;第3部分提供了较为详细的C语言相关附录。

本书既可作为计算机和非计算机专业的程序设计基础课程的教科书,又可作为工程技术人员的参考书,同时也适用于自学读者的学习与提高。

图书在版编目(CIP)数据

C语言程序设计教程/张毅坤,曹锰,张亚玲等编著.
—西安:西安交通大学出版社,2003.4
(21世纪大学计算机基础课程教材)
ISBN 7-5605-1682-3

I. C… II. ①张… ②曹… ③张… III. C语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆CIP数据核字(2003)第024193号

书 名: C语言程序设计教程
编 著: 张毅坤 曹锰 张亚玲 等
责任编辑: 贺峰涛 屈晓燕
出版发行: 西安交通大学出版社
地 址: 西安市兴庆南路25号 (邮编:710049)
网 址: <http://unit.xjtu.edu.cn/unit/jtupress>
电 话: (029)82668357 82667874(发行部)
(029)82668315 82669096(总编办)
电子邮箱: eibooks@163.com
印 刷: 陕西宝石兰印务有限责任公司
版 次: 2003年4月第1版 2003年12月第2次印刷
开 本: 787mm×1092mm 1/16
印 张: 20.5
字 数: 501千字
书 号: ISBN 7-5605-1682-3/TP·328
定 价: 26.00元

版权所有 侵权必究

西安交通大学出版社

· 安 西 ·

前言

C 语言以其独特之处而风靡世界。它集低级语言与高级语言于一体,既有低级语言可直接访问内存地址、能进行位操作、生成目标代码质量高、程序运行效率高的优点,又具有高级语言描述问题方便、运算符和数据结构丰富、结构化控制语句功能强、简洁紧凑灵活、可移植性好的优势,被人们称之为“高级语言中的低级语言”,或称之为“中级语言”。它是结构化程序设计的理想语言,符合现代编程风格,特别适合编写操作系统、网络软件和编译器等系统软件。

但是,正是由于 C 语言的丰富性、灵活性及富有特色,使得初学者感到学习困难、容易出错和难以理解。本书将本着循序渐进、由浅入深、实用高效、通俗易懂,既注重主干基础知识的流畅传送,又注重难点重点知识的分解与综合运用为指导思想,不仅表述了 C 语言的基本概念、基本规则,而且从软件工程的角度出发,阐述了分析问题、解决问题的思路和方法,并且给出了运用 C 语言实现规模较大的工程程序的实例。

作者根据自己多年的教学经验和对初学者学习、理解编程语言规律的了解,考虑到教学过程的方便性,基本知识的完整性,以及深入学习的可扩展性,将全书精心编排为三大部分。

第 1 部分为基础篇,分为 8 章,分别讲述了 C 语言的概述、数据类型与表达式、基本结构程序设计、数组、函数、结构体与共用体、指针和文件等知识的基本概念、基本规则与基本内容。力求以提出问题、目标与用途、实现的基本法则与方法、由浅入深且贯穿始终的例题为编写主线,使章与章、节与节之间逐步深入,丝丝相扣,不断扩展。不苛求面面俱到,但要做到通俗易懂。对于可讲可不讲的内容,不便于理解的内容,灵活多变、导致初学者容易出错的内容,以及对于深入学习者知识的扩展与综合运用的内容均放在本书第 2 部分讲授。同时,该篇中的一部分习题选自历年来的计算机等级考试典型试题,为部分读者备战等级考试提供了方便。

第2部分为综合扩展篇,分为4章。其中第9章重点讲述了变量的存储类别,某些运算符的结合性所引起的二义性与副作用,逗号表达式和for循环语句的灵活表述,位运算,以及编译与预处理,多文件互连等知识;第10章扩展了在Turbo C环境下,用C语言实现图形和动画的相关基本内容,给出了实现点、线、图形、色彩控制与填充、图形方式下的文本显示,以及视口操作函数等知识的基本实现方法和例子;第11章和第12章分别通过“上位机监测系统软件设计”和“超市库存货品信息管理系统设计”的两个实例,从软件工程的角度出发,介绍了问题定义、需求分析、总体设计、详细设计、编码、测试以及维护等软件开发基本步骤,并给出了用C语言实现各功能模块的源程序代码及注释,充分体现了C语言知识的综合运用,如:各种控制语句、数组、函数、结构体、指针、文件、图形等的综合运用。同时,还通过这两个实例引入了一些新的知识和扩展应用,如:动态申请、释放内存空间,用指针实现动态链表数据结构的存储、排序、插入、删除等操作,扩展了深入学习的空间。

第3部分为附录部分,分别给出了常用字符与ASCII码对照表,运算符的功能、优先级及结合性,以及较为详细的C语言标准库函数的功能、参数、库类等列表,以供参考。

本书中的程序例题均经过Turbo C或VC++ 6.0运行环境的调试,程序代码均有较为详细的注释。另外,还有《C语言程序设计教程学习指南与实验指导》一书与本书配套。该书指出了《C语言程序设计教程》学习过程中的难点和易错点,给出了解题实例和部分习题的解题思路与程序框图和答案,同时,还提出了基本实验要求。如有需求,作者还可免费提供《C语言程序设计教程》授课的电子教案。

本书由张毅坤任主编,曹锰、张亚玲任副主编。第1,2章由张毅坤编写,第4,7章由曹锰编写,第3,5,8章由张亚玲编写,第6,11,12章由王战敏编写,第9,10章由马维纲编写,最后由张毅坤统稿。西安交通大学的冯博琴教授、西北工业大学李伟华教授、西北大学的耿国华教授、西北农林科技大学的何东健教授等对本书的编写大纲及内容均提出了宝贵的意见和建议,主审冯博琴教授仔细地审阅了全书并提出了许多建设性的修改意见,同时本书的出版也得到了西安交通大学出版社的大力支持,作者在此一并向他们表示衷心的感谢。

由于作者的水平有限,本教材编写中难免有疏漏之处,恳请同行和读者不吝指正。

作者

2002年12月

目 录

第 1 部分 基础篇

第 1 章 概述

1.1 程序与程序设计语言	(3)
1.1.1 程序	(3)
1.1.2 程序设计语言	(4)
1.2 C 程序设计语言入门	(5)
1.2.1 C 语言的发展史	(5)
1.2.2 C 语言程序组成简介	(6)
1.2.3 C 程序从开发到执行的过程	(9)
1.3 C 语言的特点	(10)
习 题	(11)

第 2 章 基本数据类型、运算符及表达式

2.1 基本数据类型	(12)
2.1.1 C 的数据类型	(12)
2.1.2 C 的基本数据类型	(13)
2.2 常量	(14)
2.2.1 整型、实型及符号常量	(14)
2.2.2 字符型常量	(15)
2.3 变量	(17)
2.3.1 变量的名字	(17)
2.3.2 变量的定义	(17)
2.3.3 变量的值	(18)
2.4 运算符与表达式	(18)
2.4.1 C 运算符概述	(19)
2.4.2 算术运算符与算术表达式	(20)
2.4.3 表达式中数据间的混合运算与类型转换	(21)
2.4.4 赋值运算符与赋值表达式	(24)

小 结	(25)
习 题	(26)

第3章 基本结构程序设计

3.1 程序基本结构与结构化程序设计	(28)
3.1.1 算法	(28)
3.1.2 算法的表示	(31)
3.1.3 程序基本结构	(31)
3.1.4 结构化程序设计方法	(33)
3.2 顺序结构程序设计	(35)
3.2.1 顺序执行语句概述	(35)
3.2.2 数据的输入输出	(36)
3.2.3 顺序结构程序举例	(43)
3.3 选择结构	(44)
3.3.1 关系运算和逻辑运算	(44)
3.3.2 if 语句	(46)
3.3.3 if 语句的嵌套	(49)
3.3.4 条件运算符与条件表达式	(51)
3.3.5 switch 语句	(51)
3.4 循环结构程序设计	(53)
3.4.1 while 语句	(53)
3.4.2 do-while 语句	(55)
3.4.3 for 语句	(58)
3.4.4 循环嵌套	(59)
3.4.5 break 语句和 continue 语句	(60)
3.5 程序设计举例	(62)
小 结	(64)
习 题	(65)

第4章 数组

4.1 一维数组	(67)
4.1.1 一维数组的定义和初始化	(67)
4.1.2 一维数组元素的引用	(69)
4.1.3 应用举例	(69)
4.2 二维数组	(73)

4.2.1 二维数组的定义	(73)
4.2.2 二维数组元素的引用	(74)
4.2.3 程序举例	(75)
4.3 字符数组	(78)
4.3.1 字符数组的定义和引用	(78)
4.3.2 字符数组的输入输出	(80)
4.3.3 字符串处理函数	(81)
4.3.4 字符数组举例	(84)
小 结	(86)
习 题	(87)

第5章 函数

5.1 概述	(91)
5.1.1 C 程序的结构	(91)
5.1.2 函数分类	(92)
5.2 函数的定义与调用	(93)
5.2.1 函数定义	(93)
5.2.2 函数调用	(95)
5.2.3 函数应用举例	(99)
5.3 变量作用域	(103)
5.3.1 局部变量	(103)
5.3.2 全局变量	(105)
5.4 函数的嵌套与递归	(107)
5.4.1 函数的嵌套调用	(107)
5.4.2 函数的递归调用	(110)
小 结	(114)
习 题	(114)

第6章 结构体与共用体

6.1 结构体概述	(118)
6.1.1 结构体的引入	(118)
6.1.2 结构体类型的定义	(118)
6.2 结构体变量	(120)
6.2.1 结构体变量的定义与初始化	(120)
6.2.2 结构体变量的引用	(122)
6.2.3 结构体变量作为函数参数	(124)

6.3 结构体数组	(126)
6.3.1 结构体数组的定义与初始化	(126)
6.3.2 结构体数组元素的引用	(127)
6.3.3 结构体数组作函数参数	(128)
6.4 共用体	(130)
6.4.1 共用体类型定义	(131)
6.4.2 共用体变量定义与引用	(131)
6.4.3 应用举例	(132)
6.5 枚举类型	(135)
6.6 用 typedef 定义类型	(137)
6.6.1 类型定义的含义及形式	(137)
6.6.2 类型定义的优点	(138)
小 结	(138)
习 题	(139)

第7章 指针

7.1 地址和指针的概念	(142)
7.1.1 变量的地址和变量的值	(142)
7.1.2 直接访问和间接访问	(143)
7.1.3 指针的概念	(143)
7.2 指针变量的定义与引用	(144)
7.2.1 指针变量的定义和初始化	(144)
7.2.2 指针变量的引用	(145)
7.2.3 函数中用指针变量作形参实现变量的引用传递	(146)
7.3 指针与数组	(149)
7.3.1 指向一维数组元素的指针	(149)
7.3.2 指向多维数组元素和指向分数组的指针	(155)
7.3.3 用字符数组和字符型指针访问字符串	(158)
7.3.4 指针数组和指向指针的指针	(163)
7.4 指针与结构体	(168)
7.4.1 指向结构体变量的指针变量	(168)
7.4.2 指向结构体数组元素的指针	(169)
7.4.3 用指向结构体的指针作函数参数	(170)
7.5 指针与函数	(172)
7.5.1 返回指针值的函数	(172)

7.5.2 指向函数的指针和函数参数	(173)
小 结	(176)
习 题	(178)

第8章 文件

8.1 文件概述	(182)
8.1.1 文件的概念及分类	(182)
8.1.2 文件的操作流程	(184)
8.1.3 文件缓冲区	(184)
8.1.4 文件指针	(185)
8.2 文件的打开与关闭	(185)
8.2.1 文件的打开(fopen 函数)	(185)
8.2.2 文件的关闭(fclose 函数)	(187)
8.3 文件的顺序读写	(188)
8.3.1 字符读写函数 fgetc 和 fputc	(188)
8.3.2 字符串读写函数 fgets 和 fputs	(191)
8.3.3 格式化读写函数 fscanf 和 fprintf	(193)
8.4 文件的随机读写	(195)
8.4.1 文件定位	(195)
8.4.2 文件的随机读写	(196)
小 结	(200)
习 题	(200)

第2部分 综合扩展篇

第9章 C语言基础知识进阶

9.1 C语言基本词法进阶	(205)
9.1.1 存储类型修饰符	(205)
9.1.2 逗号表达式及灵活的 for 循环形式	(208)
9.1.3 运算符的结合性及其副作用	(211)
9.2 位运算	(213)
9.2.1 位段	(213)
9.2.2 位运算和位运算符	(215)
9.3 编译及预处理	(218)
9.3.1 文件包含 #include	(218)
9.3.2 宏定义 #define	(221)
9.3.3 条件编译	(224)

第 10 章 Turbo C 图形程序设计

10.1 基本概念、结构和函数	(227)
10.2 图形模式的初始化和检测函数	(228)
10.3 屏幕颜色的设置和清屏函数	(230)
10.4 基本的图形函数	(231)
10.4.1 点和坐标位置函数	(231)
10.4.2 画线函数	(233)
10.4.3 设定线型函数	(234)
10.5 封闭图形的填充	(236)
10.5.1 基本图形轮廓	(236)
10.5.2 设定填充方式	(236)
10.5.3 任意封闭图形的填充	(238)
10.6 图形模式下的文本输出	(240)
10.6.1 文本字符的输出	(240)
10.6.2 字体、字型和输出方式设置	(241)
10.6.3 文本字符大小设置	(243)
10.7 视口和视口函数	(246)
10.8 屏幕操作	(248)

第 11 章 综合实例一

——上位机监测系统软件设计

11.1 问题定义	(255)
11.1.1 问题背景	(255)
11.1.2 用户目标	(255)
11.2 概要设计	(255)
11.2.1 方案确定	(256)
11.2.2 软件结构	(256)
11.2.3 模块功能说明	(257)
11.3 详细设计	(257)
11.3.1 数据设计	(257)
11.3.2 流程图	(257)
11.4 编码	(259)
11.4.1 各功能模块编码	(259)
11.4.2 主程序编码	(260)
11.5 测试	(263)

第 12 章 综合实例二

——超市库存货品信息管理系统设计

12.1 问题定义	(267)
12.1.1 问题的背景	(267)
12.1.2 用户的目标	(268)
12.2 概要设计	(268)
12.2.1 方案确定	(268)
12.2.2 软件结构	(268)
12.2.3 模块功能说明	(268)
12.3 详细设计	(269)
12.3.1 数据设计	(269)
12.3.2 流程图	(271)
12.4 编码	(279)
12.4.1 各功能模块编码	(279)
12.4.2 主程序编码	(287)

第 3 部分 附录

附录 I 常用字符与 ASCII 代码对照表	(293)
附录 II 运算符和结合性	(294)
附录 III C 库函数	(295)
参考文献	(314)

第1部分

基础篇

- ➔ 第1章 概述
- ➔ 第2章 基本数据类型、运算符及表达式
- ➔ 第3章 基本结构程序设计
- ➔ 第4章 数组
- ➔ 第5章 函数
- ➔ 第6章 结构体与共用体
- ➔ 第7章 指针
- ➔ 第8章 文件

第1章 概述

随着人类进入 21 世纪信息化社会,计算机在各个领域中的应用愈来愈起到重要的作用,而其本身的科学与技术也在日新月异地迅猛发展。众所周知,计算机是由硬件系统和软件系统两大部分组成。硬件系统随着新型半导体材料的应用,以及大规模集成电路技术的不断更新,从 1946 年第一代计算机开始到目前正在研制的第五代计算机,正在以超越莫尔定律(Moore' Law)的速度飞速发展,而其体积与成本则大幅度地下降。软件系统(无论是操作系统还是应用软件)也随着硬件平台的不断提升、开发技术与手段的不断改进、应用领域的不断拓展,正在越来越广泛、深入、细致地得以开发与应用。硬件与软件是相辅相成、缺一不可的,没有软件控制,硬件系统是一堆废铁,而没有硬件平台支撑的软件,则一事无成。然而,要编制完成一定功能的软件,还必须使用一种或多种编程语言来实现。因此,本书将当前较为流行且倍受关注的程序设计语言之一——C 语言介绍给读者,为大家今后开发软件或更进一步学习其他程序设计语言打下一个良好的基础。

1.1 程序与程序设计语言

软件是计算机系统必不可少的一部分。那么,什么是软件?什么是程序?程序设计语言是如何分类的呢?本节将给出一个解答。

1.1.1 程序

什么是软件?很难对这个名词给出精确的定义。以前人们认为软件就是程序。但随着软件危机的产生,大家对软件的认识逐渐深刻而清晰起来。目前较为公认的定义是由著名的软件工程专家 B. W. Boehm 指出的:软件是程序以及开发、使用和维护所需要的所有文档。国标“软件工程技术语”中定义:“软件是与计算机系统的操作有关的计算机程序、规程、规则,以及可能有的文件、文档及数据”。由此可以看出,程序只是完整软件产品的一个部分。

国标中规定:“计算机程序是按照具体要求产生的适合于计算机处理的指令序列”。也就是说程序是为完成某一特定功能,由编程人员指定的、控制计算机按顺序执行一系列动作的、计算机能够识别的指令集合体。或者说,程序是计算机可以识别和执行的表示的处理步骤。我国颁布的《计算机软件保护条例》对程序的概念给出了更为精确的描述:“计算机程序是指为了得到某种结果而可以由计算机等具有信息处理能力的装置执行的代码化指令序列,或者可被自动地转换成代码化指令序列的符号化序列,或者符号化语句序列。”这就是说,程序要有目的性和可执行性。程序就其表现形式而言,可以是机器能够直接执行的、代码化的指令序列,也可以是机器虽然不能直接执行,但是可以转化为机器可以直接执行的符号化指令序列或符号化语句序列。

因此,程序是由某种程序设计语言编制出来,体现了编程者的控制思想和对计算机执行操作的要求。不同的任务功能,就会需求不同的软件程序,如:控制计算机本身软硬件协调工作,并使其设备充分发挥效力,方便用户使用的系统软件程序,称为操作系统;而为办公自动化(OA)、管理信息系统(MIS)、生产过程控制、计算机辅助设计(CAD)、计算机辅助制造(CAM)、人工智能、电子商务、网络互联等等应用而开发的软件程序,统称为应用软件。

1.1.2 程序设计语言

用于书写计算机程序所使用的语言称为程序设计语言。它是由人工设计的语言,是人与计算机之间交互的工具。它的好坏不仅关系到书写程序是否方便易读,而且影响到程序的质量。

程序设计语言按照书写形式以及思维方式的不同,一般分为低级语言和高级语言两大类。

低级语言包括机器语言和汇编语言。

1. 机器语言

机器语言是以二进制代码形式表示的机器基本指令的集合,是计算机系统惟一不需要翻译可以直接识别和执行的程序设计语言。它的特点是运算速度快,每条指令均为 0 和 1 的代码串,指令代码包括操作码与操作对象。如:在某一计算机中,1011011000000000 这条指令的作用是让计算机进行一次加法。但它的缺点是,机器语言随计算机机型的不同而不同,难阅读、难查错、难修改。通常,只有当编程者对 CPU 指令系统比较熟悉、需要编写的程序较短时,才有可能直接用机器语言来编写程序。人们为了摆脱编程中这种原始而低级的状态,设法采用一组字母、数字或字符来代替机器指令,这样就产生了汇编语言的概念和方法。

2. 汇编语言

与机器语言相比,使用汇编语言来编写程序可以用助记符来表示指令的操作码和操作对象,也可以用标号和符号来代替地址、常量和变量。如:ADD AX,BX;代表两个寄存器数相加的功能,这种方式便于识别与记忆,执行效率也较高。然而用汇编语言编写的程序不能由计算机直接执行,它必须通过一种具有“翻译”功能的系统支持程序——汇编程序的帮助,将这种符号化语言转换成相应的机器可执行代码,才能被执行。此外,不同 CPU 的指令系统其相应的汇编语言不同。如:单板机、单片机、微处理器等,随机器型号、类型的不同,各自的汇编语言也不同。

低级语言虽工作效率高,程序逻辑代码量小,但与人们思考问题和描述问题的方法相距甚远,使用繁琐、费时、易出差错,要求使用者熟悉计算机内部细节,非专业的普通用户很难使用。

因此,为了方便使用,程序设计语言朝着接近于人们熟悉、习惯的自然语言和数学语言描述的高级化方向发展,形成了各式各样、丰富多彩的程序设计高级语言。

高级语言包括各种面向过程和面向对象的程序设计语言。

3. 高级语言

高级语言的出现是计算机编程语言的一大进步。它屏蔽了机器的细节,提高了语言的抽象层次,程序中可以采用具有一定含义的数据命名和容易理解的执行语句。这些接近于自然语言和数学语言的语句,易学、易用、易维护,且在一定程度上与机器无关,给编程带来了极大方便。

针对不同的应用场合和用途,人们曾设计出几百种高级语言,但经过严酷的优胜劣汰过程,当今较为通用的高级语言也不过几十种。如:描述问题求解过程——面向过程的设计语言典型的有:适用于科学计算的 FORTRAN 语言;适用于商业事务处理的 COBOL 语言;适用于初学者的 BASIC 语言;第一个系统体现结构化程序设计思想的教学工具 PASCAL 语言;功能丰富,移植性强,编译质量高的 C 语言;以及用于人工智能程序设计的 PROLOG 语言等等。而着眼于问题域的对象及其相互关系,更加符合人类对客观事物认识过程及思维方式的面向对象程序设计语言典型的有:第一个真正面向对象的程序设计语言 Smalltalk 语言;嵌入式现代模块化语言 Ada 语言;对 C 语言进行革命性扩充与改进的 C++ 语言;以及现代流行的 Java, Visual Basic, Visual C++, Delphi 等程序设计语言。

但是,由于计算机硬件不能直接识别高级语言中的语句,因此,必须经过“翻译程序”,将用高级语言编写的程序翻译成计算机硬件所能识别的机器语言程序方可执行。所以说,用高级语言进行程序设计,其编程效率高,方便易用,但执行速度没有低级语言高。而且针对每一种高级语言的本身设计,既要能满足编程人员的需求,很方便、自然地描述现实世界中的问题,又要能构造出高效的编译程序,能够把程序设计语言的所有内容翻译成高效的机器指令,也不是一件容易的事情。

1.2 C 程序设计语言入门

C 程序设计语言是当前功能最为强大、国际上流行最为广泛、适用范围最宽的高级程序设计语言之一。

1.2.1 C 语言的发展史

C 语言是从 BCPL(Basic Combined Programming Language)语言和 B 语言(取 BCPL 的第一个字母)演化而来的。BCPL 是 1967 年由 Mantin Richards 为编写操作系统软件和编译器而开发的语言。Kem Thompson 在模拟了 BCPL 语言的许多特点的基础上开发出了简单且很接近硬件的 B 语言,并于 1970 年在贝尔实验室用 B 语言在一台 DEC PDP-7 计算机上实现了第一个 UNIX 操作系统。BCPL 和 B 语言都是“数据无类型”语言,即每一个数据项都占用内存中的一个字,处理数据项的责任落在程序员身上。

C 语言(取 BCPL 的第二个字母)是贝尔实验室的 Dennis M. Ritchie 在 B 语言的基础上开发出来的。1972 年在一台 DEC PDP-11 计算机上实现了最初的 C 语言。C 语言既保持了 BCPL 和 B 语言的优点(精练、接近硬件),又克服了它们的缺点(过于简单、数据无类型等),增加了数据类型和其他强大功能,使它作为 UNIX 操作系统的开发语言而开始广为人知。实际