



微型计算机基础知识教育丛书（新版）

C语言 基础教程

（修订版）

吕凤翥 编著

北京大学出版社

PEKING UNIVERSITY PRESS



微型计算机基础知识教育丛书(新版)

C 语言基础教程

(修订版)

吕凤翥 编著

北京大学出版社
北 京

内 容 简 介

本书作者总结了十多年来专心从事C语言教学的经验,全面系统地讲解了C语言的基本语法和基础的编程方法。本书包含有词法和词法规则,常量、变量,运算符和表达式,语句,函数和存储类,预处理功能,指针,结构,联合和枚举,以及文件操作等内容。本书在讲述上具有突出重点、详述难点、揭示疑点的特点。有近150个不同类型的例题,每章都有丰富的思考题和作业题。本书语言通俗、概念准确、抓住读者心理,回答读者问题,适于自学。

本书可作为高等院校本科和大专学生的教材和教员的教学参考书,也可作为成人教育、自学考试教材和参考书。

图书在版编目(CIP)数据

C语言基础教程/吕凤翥编著. —北京:北京大学出版社,1998.2
(微机计算机基础知识教育丛书)

ISBN 7-301-03668-X

I. C… I. 吕… III. C语言-程序设计-基本知识 N. TP312

书 名: C语言基础教程(修订版)

著作责任者: 吕凤翥

责任编辑: 杨锡林

标准书号: ISBN 7-301-03668-X/TP·391

出版者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

电 话: 出版部 62752015 发行部 62559712 编辑部 62752032

排 印 者: 北京经纬印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787×1092 16开本 18.25印张 450千字

1998年3月第一版 1998年3月第一次印刷

定 价: 26.00元

序 言

从第一台电子计算机问世到今天,几近半个世纪,人类从生产到生活发生了巨大的变化,电脑已悄然闯入社会生活的各个领域。过去说:没有电将寸步难行;现在要说:没有计算机就没有现代化。

计算机科学是信息科学的一个重要组成部分。21 世纪将以信息技术为主导,使整个社会的经济活动方式与社会的就业结构产生非常大的变化。体力劳动的比重将逐渐减少,掌握信息技术的脑力劳动者的比例将不断增大。电子函件、电子新闻、电子图书等新的科技将逐步取代纸笔和印刷机,新的计算机文化将迅速发展。

著名的计算机科学家 G·伏赛斯曾预言:电脑将是继自然语言、数学之后而成为第三位的,对人的一生都有大用处的“通用智力工具”。现在,实践已经证明了电脑已经成为各行各业的基本工具。许多部门已经把具备电脑的应用知识与技能作为录用或考核工作人员的一个重要条件。综合国力的竞争说到底还是掌握高科技人才的竞争。怎样将计算机科学知识迅速而有效地普及到全社会,也就成了一件具有紧迫感的新任务。

近年来为适应社会的需求,各类职业教育学校有了较快的发展。在这些学校里的学生理所当然地要接受计算机教育。但是,目前的状况是,适用于这些学校的教材却非常之少。因此,尽快写出这种教材供同学们选用,是我们编写“微型计算机基础知识教育丛书”的初衷。从教学目标出发,这套丛书将重点讲述基本概念和基本方法,以理论联系实际思路介绍一些具体的实际操作技术;在写作手法上,力求通俗而不肤浅,深入而不玄奥,贯彻循序渐进的原则;在每一应知应会的知识点上,着力讲深讲透;书中附有必要的思考题和上机练习题,引导读者既动脑又动手,学深,学活,学以致用。

随着电脑应用的普及,蒙在电脑上的一块神秘的面纱已经被揭落。许多学过电脑的人都感到,入门不难,深造也是办得到的。只要功夫深,电脑不会不听命。

中国计算机学会普及委员会主任
清华大学计算机科学与技术系教授

吴文虎

1997.3.25

前 言

本书是在《实用C语言基础教程》和《C语言入门》(这两本书都由北京大学出版社出版)的基础上,根据广大读者的需要,总结了十多年来从事C语言教学的经验编写而成的。

本书在内容上和编排上都作了较大的调整,较明显地体现出突出重点、详述难点、揭示疑点的特点。本书经修改后,内容上更全面和系统,它包含了C语言所有的基础知识和基本技能;讲解上更加深入细致,先讲清概念,指出方法和规则,然后提醒在实际应用中的注意事项,最后通过列举例题加深理解和学会应用,对每个例题都做了详尽的分析,并指出编程的技巧和方法。本书的例题增加了,全书近150个不同类型的例题,每个例题都具有一定的代表性,有的是训练语法的,有的是练习编程的,从每个例题中都会得到收获;作业题的内容更加丰富了,有判断、有填空、还有分析程序结果和编程,既可训练对基础知识的理解,又可学会分析问题、解决问题的方法,提高C语言的编程能力。

作者根据十多年来从事C语言教学的经验,较好地抓住读者学习过程中的问题,并在书中予以解答。总之,修订后的本书更适合作为教材和自学指导书。

本书共分十章。由浅入深系统地全面地讲述了C语言的语法知识和编程技术。前三章讲述了C语言的发展、特点和应用,C语言的词法及其规则,C语言编程特点;还讲述了C语言的基础知识:常量、变量、类型转换、运算符和表达式等内容,这里强调了运算符的种类、功能、使用方法、优先级和结合性等,这是一个重点和难点。第四至六章讲述了C语言中主要的语法内容:语句和函数,同时讲解了存储类和预处理功能;这部分重点讲解函数的定义和说明,函数参数和返回值以及函数的调用,突出两种不同形式的传值调用;这部分难点是变量的存储类和各类标识符的作用域。第七章指针是C语言的最大特点,指针也是学习C语言的最大难点,本书用较大篇幅详尽地讲述了指针的概念和指针数组和函数方面的具体应用;强调在实际应用中理解指针概念和使用指针的方法;本章通过大量例题展示出指针的广泛应用,从中掌握指针的本质及运算规则,这对学会使用指针有极大帮助。第八、九两章,讲述了C语言的三种构造的数据类型:结构、联合和枚举;这里,结构是重点;通过讲述结构的概念和应用,学会结构的使用方法,并看到结构在解决实际问题中的重要性。最后一章讲述了文件的操作,它包括C语言文件的特点,标准文件的读写函数的操作方法、一般文件的打开和关闭操作、读写操作和定位操作等。

本书每章后面都有复习本章内容的思考题和检验对本章内容理解和掌握程度的作业题,作业题中包含有训练概念的填空和判断题,练习操作方法的分析结果题和培养编程能力的编程题。作业题中的分析程序结果题和编程题可以上机验证,从而得知你的分析是否正确和你编写的程序是否符合要求。这些题目不仅种类较多,而且类型也较全,重复性很小,不仅适用读者练习,也可用于教员选取作为学生的作业题,或稍加变化可作为考试题。

当你读完本书后,可能对C语言产生了浓厚的兴趣,还想继续使用C语言进行程序设计的方法,你可以继续读本人编写的《C语言应用教程》(该书已由北京大学出版社出版),它会帮助你学会C语言的编程方法,同时会看到C语言的运用范围可真广啊!

本书的所有程序(例题和作业题中程序)都在 Turbo C 语言编译系统下调试过,并备有程序软盘。

由于时间较紧,难免会有错误存在,请读者指正。

谢谢喜欢我编著的计算机书籍的读者。

作 者

1997 年 10 月于北大燕北园

目 录

第一章 C语言概述	(1)
1.1 C语言的由来与发展	(1)
1.1.1 C语言的由来	(1)
1.1.2 C语言的发展	(1)
1.2 C语言的特点和应用	(2)
1.2.1 C语言的特点	(2)
1.2.2 C语言的应用	(5)
1.3 C语言的词法及其规则	(6)
1.3.1 字符集	(6)
1.3.2 单词及词法规则	(6)
1.4 C语言常用的输入输出函数	(9)
1.4.1 常用的输入函数	(9)
1.4.2 常用的输出函数	(10)
1.5 C语言程序实例及其实现	(12)
1.5.1 C语言程序实例	(12)
1.5.2 C语言程序书写格式	(14)
1.5.3 C语言程序实现	(15)
练习题	(17)
作业题	(17)
第二章 常量、变量和类型转换	(19)
2.1 常量	(19)
2.1.1 数字常量	(19)
2.1.2 字符常量和字符串常量	(20)
2.1.3 符号常量	(22)
2.2 变量	(23)
2.2.1 变量的名字	(23)
2.2.2 变量的类型	(24)
2.2.3 变量的值	(25)
2.3 数组	(27)
2.3.1 数组的定义	(27)
2.3.2 数组的赋值	(28)
2.3.3 字符数组	(31)
2.4 类型转换	(34)
2.4.1 自动转换	(34)
2.4.2 强制转换	(35)
练习题	(35)

作业题	(36)
第三章 运算符和表达式	(38)
3.1 常用运算符的功能	(38)
3.1.1 算术运算符	(38)
3.1.2 增 1 减 1 运算符	(39)
3.1.3 关系运算符	(40)
3.1.4 逻辑运算符	(40)
3.1.5 位操作运算符	(41)
3.1.6 赋值运算符	(42)
3.1.7 其他运算符	(43)
3.2 运算符的优先级和结合性	(45)
3.2.1 运算符的优先级	(45)
3.2.2 运算符的结合性	(45)
3.3 表达式	(47)
3.3.1 表达式和表达式的种类	(47)
3.3.2 表达式的值和类型	(56)
3.3.3 表达式求值中值得注意的两个问题	(57)
练习题	(60)
作业题	(61)
第四章 语句	(63)
4.1 表达式语句和空语句	(63)
4.1.1 表达式语句	(63)
4.1.2 空语句	(64)
4.2 复合语句和分程序	(64)
4.2.1 复合语句	(64)
4.2.2 分程序	(65)
4.3 分支语句	(65)
4.3.1 条件语句	(65)
4.3.2 开关语句	(70)
4.4 循环语句	(76)
4.4.1 while 循环语句	(76)
4.4.2 do-while 循环语句	(77)
4.4.3 for 循环语句	(79)
4.5 转向语句	(85)
4.5.1 goto 语句	(85)
4.5.2 break 语句	(86)
4.5.3 continue 语句	(87)
4.5.4 return 语句	(88)
练习题	(89)
作业题	(89)
第五章 函数和存储类	(95)

5.1	函数的定义和说明	(95)
5.1.1	函数的定义	(95)
5.1.2	函数的说明	(97)
5.2	函数的参数和返回值	(97)
5.2.1	函数的参数	(97)
5.2.2	函数的返回值	(99)
5.3	函数的调用	(100)
5.3.1	传值调用的特点	(100)
5.3.2	传址调用的特点	(101)
5.3.3	数组名作参数的函数调用	(103)
5.3.4	函数的嵌套调用	(105)
5.3.5	函数的递归调用	(106)
5.4	作用域规则	(110)
5.4.1	标识符的作用域规则	(110)
5.4.2	重新定义变量的作用域规定	(111)
5.5	存储类	(112)
5.5.1	变量的存储类	(112)
5.5.2	函数的存储类	(118)
	练习题	(122)
	作业题	(122)
第六章	预处理功能和类型定义	(129)
6.1	预处理功能概述	(129)
6.2	宏定义	(129)
6.2.1	简单宏定义	(130)
6.2.2	带参数的宏定义	(132)
6.2.3	宏定义的应用	(135)
6.3	文件包含	(136)
6.3.1	文件包含命令的格式和功能	(136)
6.3.2	使用文件包含命令时应注意事项	(137)
6.4	条件编译	(139)
6.4.1	条件编译的常用命令格式	(139)
6.4.2	条件编译命令的应用	(140)
6.5	类型定义	(142)
6.5.1	类型定义的含意和类型定义语句	(142)
6.5.2	类型定义的应用	(144)
	练习题	(144)
	作业题	(145)
第七章	指针	(150)
7.1	指针的概念	(150)
7.1.1	什么是指针	(150)
7.1.2	指针的表示	(151)

7.1.3	指针的赋值	(153)
7.1.4	指针所指向变量的值	(155)
7.2	指针的运算	(157)
7.2.1	指针的赋值运算	(157)
7.2.2	指针加减整数的运算	(157)
7.2.3	两个指针相减的运算	(157)
7.2.4	两个指针比较的运算	(158)
7.2.5	指针运算与地址运算的区别	(159)
7.3	指针与数组	(160)
7.3.1	数组名是一个常量指针	(160)
7.3.2	数组元素的指针表示	(161)
7.3.3	字符数组、字符指针和字符串处理函数	(168)
7.3.4	指向数组的指针和指针数组	(173)
7.4	指针与函数	(179)
7.4.1	指针作函数参数	(180)
7.4.2	指针函数和指向函数的指针	(185)
	练习题	(190)
	作业题	(190)
第八章	结构	(198)
8.1	结构的概念	(198)
8.1.1	结构和结构变量的定义	(198)
8.1.2	结构变量成员的表示	(200)
8.1.3	结构变量的赋值	(201)
8.1.4	结构变量的运算	(203)
8.2	结构与数组	(204)
8.2.1	数组与结构成员	(204)
8.2.2	结构数组	(205)
8.3	结构与函数	(209)
8.3.1	结构变量与指向结构变量的指针作函数参数	(209)
8.3.2	结构变量和指向结构变量的指针作函数返回值	(212)
8.4	链表	(213)
8.4.1	链表的概念	(213)
8.4.2	链表的操作	(214)
8.5	位段	(224)
8.5.1	位段的概念	(224)
8.5.2	使用位段时应注意的事项	(226)
	练习题	(227)
	作业题	(227)
第九章	联合和枚举	(233)
9.1	联合的概念	(233)
9.1.1	联合变量的定义和赋值	(233)

9.1.2 联合与结构的区别	(234)
9.2 联合的应用	(236)
9.3 枚举的概念	(239)
9.3.1 枚举变量的定义和赋值	(239)
9.3.2 使用枚举变量时应注意的事项	(240)
9.4 枚举的应用	(241)
练习题	(243)
作业题	(243)
第十章 文件和读写函数	(246)
10.1 C语言中文件的概念	(246)
10.1.1 文件和文件指针	(246)
10.1.2 标准文件和一般文件	(247)
10.1.3 高级读写函数和低级读写函数	(248)
10.2 标准文件的读写操作	(248)
10.2.1 标准文件读写函数介绍	(248)
10.2.2 标准文件读写函数应用	(251)
10.3 一般文件的操作	(255)
10.3.1 打开文件函数和关闭文件函数	(255)
10.3.2 一般文件读写函数及其使用	(256)
10.3.3 文件定位函数及其使用	(263)
10.4 介绍常用的其他函数	(266)
10.4.1 动态存储分配函数	(266)
10.4.2 系统调用函数	(267)
10.4.3 字符函数	(268)
10.4.4 常用数学函数	(269)
练习题	(269)
作业题	(270)
附录	(276)
附录1 ASCII 编码表	(276)
附录2 ctype.h 文件中所包含的字符函数	(278)
附录3 math.h 文件中所包含的数学函数	(279)

第一章 C 语言概述

本章介绍 C 语言的由来和发展、C 语言的特点和应用、C 语言程序的书写格式和实现,使读者对 C 语言程序结构有一个初步的了解,并对 C 语言建立一个整体概念。本章还介绍 C 语言程序中常用的输入输出函数以及 C 语言的词法和语法规则,为后面各章的学习打下一个基础。

1.1 C 语言的由来与发展

1.1.1 C 语言的由来

C 语言诞生于 1972 年,由美国电话电报公司(AT&T)贝尔实验室的 D. M. Ritchie 设计,并首先在一台使用 UNIX 操作系统的 DEC PDP-11 计算机上实现。

C 语言是在一种称为 B 语言的基础上,克服了 B 语言依赖于机器又无数据类型等局限性开发的语言。在 1970 年,美国贝尔实验室的 K. Thompson 以 BCPL 语言为基础,设计出一种既简单又接近于硬件的 B 语言,并用它写成了第一个 UNIX 操作系统,在 PDP-7 计算机上实现的。B 语言是取了 BCPL 语言的第一个字母。而 BCPL 语言(Basic Combined Programming Language)是 1967 年英国剑桥大学的 M. Richards 基于一种 CPL 语言(Combined Programming Language)提出的一种改进的语言。而 CPL 语言又是于 1963 年英国剑桥大学根据 ALGOL 60 推出的一种接近硬件的语言。由此可见,C 语言的根源可以追溯到 ALGOL 60,它的演变过程如下所示:

ALGOL60(1960 年)→CPL(1963 年)→BCPL(1967 年)→B(1970 年)→C(1972 年)

1.1.2 C 语言的发展

C 语言是在人们设想寻找一种既具有一般高级语言的特征,又具有低级语言特点的语言的情况下应运而生的,它具有人们的这种期望,集中了高级语言和低级语言的优点。最初的 C 语言就是为了描述和实现 UNIX 操作系统而产生的一种工具语言。1973 年,贝尔实验室的 K. Thompson 和 D. M. Ritchie 两人合作使用 C 语言修改了 UNIX 操作系统,即 UNIX 第 5 版本。原来的 UNIX 操作系统是用汇编语言写的,改写后 UNIX 操作系统中 90%以上使用了 C 语言。从此,C 语言的命运与 UNIX 操作系统便有着密切的联系,随着 UNIX 操作系统的发展和推广,C 语言也在被广泛的使用和发展。

C 语言出世以后,在应用中不断的改进。在 1975 年 UNIX 第 6 版本公布以后,C 语言开始引起人们的注意,它的优点逐步被人们所认识。1977 年出现了与具体机器无关的 C 语言编译文本,推动了 UNIX 操作系统在各种机器上的迅速地实现。随着 UNIX 的日益广泛的使用,C 语言也得到了迅速的推广。1978 年以后,C 语言先后被移植到大、中、小和微型机上,它很快成为世界上应用最广泛的计算机语言之一。

1978 年又推出 UNIX 第 8 版本,以该版本中的 C 编译程序为基础,B. W. Kernighan 和

D. M. Ritchie 合作(被称为 K&R)出版了《The C Programming Language》(C 程序设计语言)一书,被称为标准 C。1983 年,ANSI(美国国家标准化协会)对 C 语言的各种版本进行了扩充,推出了新的标准,被称为 ANSI C,它比原来的标准 C 有了改进和扩充。1987 年,ANSI 又公布了 87 ANSI C 新版本。目前流行的各种 C 语言编译系统的版本大多数都以此为基础,但各有其不同。当前微机上使用的 C 语言编译系统多为 Microsoft C, Turbo C, Borland C 和 Quick C 等,它们略有差异,按标准 C 书写的程序,基本上都可运行。读者要了解不同版本的编译系统的特点和区别可参阅有关的操作说明书。

1.2 C 语言的特点和应用

1.2.1 C 语言的特点

C 语言是一种开发比较晚的高级语言,它吸取了早期高级语言的优点,克服了某些不足,形成了它独有的特性。C 语言的特点概括起来有如下几点。

1. C 语言是一种结构化的程序设计语言

结构化程序设计的优点是便于分工合作,便于调试、维护和扩充。这种程序设计方法是把一个程序分成若干个模块,每个模块完成一个功能,由一个总控模块来控制 and 协调各个模块来实现总的功能;因此,这种程序设计方法又称为模块化程序设计方法。在 C 语言中,函数是构成 C 语言程序的最小模块。实际上,C 语言的程序是由一个或者多个文件组成的,每个文件又是由一个或多个函数组成的。因此,一个程序是由一个或多个函数组成的,其中,须有且仅有一个主函数,主函数的名字规定为 `main()`。这样,组成一个程序的若干个文件中,仅有一个是主文件,只有主文件中才含有 `main()` 函数。另外,函数是由若干条语句组成的,语句是由若干个单词组成的,单词是由若干个字符组成的。字符是构成程序的最小单元。C 语言程序的构成如下所示:

程序→文件→函数→语句→单词→字符。

C 语言是结构化程序设计语言,它具有构成三种基本结构模式的语句,这种结构化程序设计应具有三种基本结构模式如下:

● 顺序结构模式

它将由若干条顺序执行的语句构成,这是程序设计的最基本形式。

● 分支结构模式

C 语言中具有条件语句和开关语句,它将会构成各种需要的分支结构模式。

● 循环结构模式

C 语言中提供了三种循环语句(`for` 循环, `while` 循环和 `do-while` 循环),用它们可以构成各种循环结构模式。

在讲述 C 语言是一种结构化程序设计语言的同时,还必须指出它与其他结构化程序设计语言(如 PASCAL 语言)相比较,还有一些不完全之处,因此,C 语言是一种不完全的结构化程序设计语言。其表现如下:

第一,完全的结构化程序设计语言不允许使用 `goto` 语句,因为 `goto` 语句会破坏结构化。但是,`goto` 语句在某些时候会使得程序简练,因此,在 C 语言中原则上允许使用 `goto` 语句,为了减少它对结构化的影响,采取了限制使用的办法,即规定 `goto` 语句的转向范围只能在一个

函数体内。不得使用 goto 语句从一个函数体内转向到另一个函数体中,这种限制性地使用 goto 语句会给编程带来一些方便,又不会影响模块之间的结构化。但是,建议在 C 语言的程序中尽量少使用 goto 语句。

第二,完全的结构化程序设计语言要求一个模块只有一个入口和一个出口,这样便于结构化的管理。但是,C 语言程序中允许函数使用多个返回语句(return 语句),即允许函数有多个出口,返回到调用函数。这样做也是为了编程中的方便。在 C 语言程序中,可以使用条件语句来返回不同的函数值。

由此可见,C 语言虽然是一个不够严格的结构化程序设计语言,但它是一个使用起来十分灵活的高级语言。

2. C 语言十分简练

C 语言是一种非常简练的语言。用 C 语言编写的程序十分简洁。C 语言的简洁性表现如下:

(1) C 语言中类型说明符采用缩写形式,例如,整型可用 int,而不用 integer;字符型用 char 而不用 character;长整型可用 long 等等。

(2) C 语言中关键字较少,只有 32 个。有些关键字用简单的符号代替,例如,条件语句中的 if 体的定界符采用花括号({}),如果是一条语句规定不用定界符。又例如,循环语句中循环体也是如此。

(3) 运算符丰富。不仅数量多,而且功能强,例如,三目运算符(?:)具有条件语句的功能。三目运算符使用方法如下:

```
d1? d2:d3
```

其中,d1,d2,d3 是不同的表达式。其功能是先计算表达式 d1 的值,如果 d1 的值是非零,则上述三目运算符组成的条件表达式的值为 d2 表达式的值;否则(即 d1 表达式的值为零)条件表达式的值为 d3 的值。因此,它将相当如下表示的 if 语句:

```
if (d1)
    c=d2;
else
    c=d3;
```

其中,c 用来存放上述条件表达式的值。

使用功能强的运算符可以使得程序简洁。

(4) 预处理功能将简化程序书写内容。C 语言中提供了一种预处理功能,它包含宏定义和文件包含等,其中宏定义有一种简化书写的功能,它可将一个复杂的格式用一个简单格式来定义,例如

```
#define PI 3.14159265
```

表示定义 PI 为 3.14159265,在程序中只需用 PI 表示,在编译前系统将用 3.14159265 来替代 PI。这里,简化是一个特点,还有其他好处后面会介绍。还有文件包含也将起到简化书写的目的,它将许多文件都将使用的一些语句放在某个指定的文件(一般用头文件,即.h 文件)中,那么在某个文件中需要这些语句时,不必重写,只要将原来写好的指定文件包含进来就可以了,其方法十分简单,例如

```
#include "myfile.h"
```

其中,myfile.h 是事先写好的需要包含其内容的文件。

3. C 语言功能很强

C 语言的功能性强表现在它既具有高级语言的功能又具有低级语言的功能。数值运算和非数值运算的功能 C 语言都具有,并且在处理非数值数据时更加方便和灵活。此外,C 语言还具备一些低级语言的功能,例如,寄存器运算功能,二进制位运算功能和内存地址运算功能等。这些低级语言的功能是一般高级语言所没有的。由于 C 语言具有上述这些功能,因此,它的应用十分广泛,它不仅像其他高级语言一样编写一些应用程序,而且还可以像汇编语言一样编写一些系统程序。而在实际上,许多系统软件,例如关系数据库管理系统,绘图软件系统等,都是用 C 语言编写的。另外,使用 C 语言编写一些接口程序也十分方便。

C 语言还提供了丰富的数据类型,除了基本的数据类型之外,还提供了构造的数据类型,例如,数组、结构、联合和枚举。使用这些数据类型可以很方便地实现各种复杂的数据结构(例如,链表、栈、树等)的操作。

4. C 语言的可移植性好

高级语言的可移植性都比汇编语言好。在诸多的高级语言中,C 语言的移植性更为突出。这是因为该语言编译系统较小,另外预处理功能对移植也带来一些方便,因此,C 语言本身只需稍加修改便可用于各种型号的机器上和各类操作系统中,用 C 语言编写的程序也很方便地用于不同系统中,这也是 C 语言得以广泛应用的原因之一。

在了解和掌握 C 语言上述的特点的同时,还应该知道 C 语言所存在的不足,这些不足往往是由于突出某个特点而带来的。在学习 C 语言中,了解这些不足是很重要的,它可以避免出现一些莫名其妙的错误。下面列举四个方面的不足。

(1) 运算符多,难用难记。C 语言中有 40 多个运算符,又分 15 种优先级和两类结合性,这无疑对数据的运算和处理带来了方便。但是,诸多的运算符和不同的优先级势必会带来难记忆,难使用的不足。例如,有些功能不同的运算符,却使用相同的符号,如“*”,它作为单目运算符表示取内容,它作为双目运算符表示两个操作数相乘,它与斜线符“/”连在一起(/ * 或 * /)表示注释符,另外,它用在说明语句中,用来表示它右边变量为指针等等。还有运算符的 15 种优先级不要记混了,否则会造成计算值的错误。

(2) C 语言中类型转换比较灵活,例如,int 型与 char 型可以自动转换。这些自动转换的规定带来一定的计算方便,在 C 语言中允许一个字符与一个整数进行加减运算,例如

```
'a'+2
```

是合法的,该表达式的值用字符型表示为 'c',用整数型表示为 99。为了转换上的方便,在许多情况下将不作类型检查,例如,在只作简单说明时的函数调用中,要求形参和实参类型一致,如果类型不一致也不判错仍可通过,有时会造成运算结果的错误,为了避免这类错误的发生,可以对函数进行原型说明,这时编译时将会对形参和实参的类型进行检查,增加其安全性。另外,也可使用强制类型运算符来限制其不必要的自动转换。

(3) C 语言中数组在动态赋值时不作越界检查。因此,在数组元素的个数少于实际赋值的项数时,编译系统不报越界错,而继续按数组元素所表示的地址进行赋值,这样容易造成数据的混乱。为了避免这种情况的发生,应尽量避免动态的越界赋值。对数组的静态赋值(即赋初值)不会发生越界赋值的现象,静态时出现越界赋值将报错。

(4) C 语言为了优化等原因允许不同的编译系统在表达式或参数表内重新安排求值顺序。这样对于一般的表达式值和参数表内各项参数值是没有影响的,例如,改变表达式中各操作数的计算顺序不会改变表达式的值。但是,对一些具有副作用的运算符(如,增 1 减 1 运算符和赋值运算符)来讲,不同的求值顺序将会造成表达式的不同的值。这一点在编写程序时应特别注意,避免出现那些可能有二义性的表达式和参数表,关于这方面详细讲解请见本书后面内容。

1.2.2 C 语言的应用

从前面对 C 语言的特点的分析中,不难看出 C 语言具有编程方便、语句简练、功能很强、移植性好等优点,它是编程者喜欢使用的一种结构化程序设计语言。

C 语言已被广泛地应用于系统软件和应用软件的开发中。在下述的几个方面应用得更为广泛。

1. 数据库管理和应用程序方面

C 语言的非数值处理功能很强,因此它被广泛地应用于数据库管理系统和应用程序。大多数的关系数据库管理系统,如 dBASE, FoxBASE, ORACLE 等,都是由 C 语言编写的。各种不同部门的应用软件也大都是用 C 语言开发的,C 语言在开发数据库应用软件方面应用很广,深受开发者的欢迎。

2. 图形图像系统的应用程序方面

C 语言在图形图像的开发中也有着广泛的市场。很多图形图像系统,如 AutoCAD 通用图形系统等,就是使用 C 语言开发的,并且在这些图形系统中可以直接使用 C 语言编程,实现某些功能。C 语言编译系统带有许多绘图功能的函数,利用这些函数开发图形应用软件十分方便。所开发的应用程序常用 C 语言编写接口界面,这样既方便又灵活,效果很好。这是因为该语言提供有图形处理功能,便于实现图形图像的各种操作。因此,C 语言在图形图像的应用方面很好地发挥了它的作用。

3. 编写与设备的接口程序方面

C 语言不仅在建立友好界面方面有着广泛应用,如下拉式菜单、弹出菜单、多窗口技术等;而且在编写与设备的接口程序方面也有着广泛应用。这是因为 C 语言不仅具有高级语言的特性还具有低级语言的功能,因此,在编写接口程序方面十分方便,有时它与汇编语言一起使用,会显示出更高的效率。

4. 数据结构方面

由于 C 语言提供了十分丰富的数据类型,不仅有基本数据类型还有构造的数据类型,如数组、结构和联合等,把它们用于较复杂的数据结构(例如,链表、队列、栈、树等)中显得十分方便,这方面已有许多成熟的例程供选择使用。

5. 排序和检索方面

排序和检索是数据处理中最常遇到并较为复杂的问题。使用 C 语言来编写排序和检索各种算法的程序既方便又简洁。特别是有些排序算法采用了递归方法进行编程,更显得清晰明了。因此,人们喜欢使用 C 语言来编写这方面的程序。

上述列举了五个方面的应用,但绝不是说 C 语言的应用仅限如此,而是说在这几个方面目前使用得更多些。C 语言可以说在各个领域中都可以使用,并且都会有较好的效果。所以,C

语言是当前被用于编程的最广泛的语言之一。

另外,C 语言是一种结构化程序设计语言,在编写大型程序中也很方便,特别是该语言又提供了预处理功能,其中文件包含在多人同时开发一个大程序时将带来减少重复和提高效率等好处,因此,越来越多的人喜欢用 C 语言来开发大型程序。

1.3 C 语言的词法及其规则

1.3.1 字符集

字符是构成 C 语言程序的最小单元,若干字符组合成单词。下面给出 C 语言中使用的合法字符。

1. 字母和数字

小写字母: a b c d e f g h i j k l m n o p q r s t u v w x y z

大写字母: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

数字: 0 1 2 3 4 5 6 7 8 9

2. 标点和特殊字符

字 符	名 称	字 符	名 称	字 符	名 称
,	逗号	{	左花括号	#	# 字号
.	点	}	右花括号	%	百分号
;	分号	<	小于号	&	和号
:	冒号	>	大于号	^	脱字符
'	单撇号	!	惊叹号	*	星号
"	双撇号		竖线	-	减号
(	左圆括号	/	斜线	=	等号
)	右圆括号	\	反斜线	+	加号
[	左方括号	~	求反号		
]	右方括号	_	下划线		

3. 空白符和空字符

空白符包含空格符、回车符、换行符、水平制表符等。

空字符是指 ASCII 码值为零的那个字符。该字符在 C 语言中有特殊用处,用它来作为字符串的结束符。

以上字符的集合称为 C 语言的合法字符集。

1.3.2 单词及词法规则

单词是由若干个有序的字符组成的,单词的集合称为词汇。C 语言的单词有如下几种:标识符、关键字、运算符、分隔符、常量、字符串和注释符。

下面对上述 7 种单词的词法规则逐一详述,有些单词,例如,关键字、运算符、常量和字符串等,后面的章节中还会详细讲述。