

● 微机用户系列丛书



语言实用教程

● 费宗莲 编

● 电子工业出版社

微机用户系列丛书

C 语言实用教程

费宗莲 编

电子工业出版社

(京)新登字 055 号

内容提要

本书全面系统地介绍了 C 语言,内容丰富,通俗易懂。全书共分 10 章。第 1 章作初步介绍;第 2 章讲述 C 语言的基本概念,第 3 章至第 9 章详细介绍 C 语言的内容,第 10 章介绍输入/输出和 UNIX 环境。每章均附有小结和习题。

C 语言是一种通用的程序设计语言,它的特点是简洁有效,经济实用,可移植性强。它不仅可用作系统程序设计语言,也同样可用作应用程序设计语言。

本书适合用作大专院校有关专业的教科书,也可作为科技人员和程序设计人员学习 C 语言的入门指导和参考书。

C 语言实用教程

费宗莲 编

责任编辑:耕耘

*

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经售

煤炭工业出版社印刷厂印刷

*

开本:850×1168 毫米 1/32 印张:13.5 字数:338 千字

1992 年 9 月第一版 1992 年 9 月第一次印刷

印数:1—12000 册 定价:9.80 元

ISBN 7-5053-1654-0/TP·351

前 言

在过去的十多年中,计算机世界发生了一场革命。个人计算机具有十年前大型机的功能。与此同时,C 语言随着 UNIX 系统的成功得到普及推广,而它又超越仅作为 UNIX 操作系统的书写语言。C 语言不仅用于书写操作系统、编译系统、数据库管理系统等系统软件,也用于数据处理、CAD、软件工程、人工智能等广阔的应用软件领域。

C 语言有很多突出的优点。作为一种通用的程序设计语言,C 语言具有表达简洁、数据结构先进、控制流和操作功能丰富等特点,使语言表现力强,有利于运用结构程序设计技术,用户编程效率得以提高。

本书全面系统地介绍 C 语言,是一本 C 语言的实用教程,适合于学生和专业科技人员使用。它可以作为学生学习程序设计的初步入门指南或者中等程度的教材,也可以结合其它一些课程,如操作系统、数据结构、图形学、数值分析、编程方法学等作为教学的参考书。本书的编写从介绍语法、语义开始,结合了基本的程序设计技术,叙述简炼,内容丰富,概念清楚,结构完整,循序渐进,通俗实用。

全书共分十章。第 1 章初步介绍 C 语言的入门知识,C 语言的特点以及如何编程和运行 C 程序。第 2 章是 C 语言的概述。介绍 C 语言的基本概念,如变量、表达式、输入/输出函数、控制流等,并开始引入函数、数组、指针的概念,使读者对 C 语言有一全

貌了解。从第 3 章开始分别逐一介绍 C 语言的内容。第 3 章描述 C 语言的基本成分:变量和常量,引入了基本数据类型:整型、字符型和浮点型。第 4 章集中讲述了 C 语言的各种运算符和表达式。第 5 章介绍控制流及相应的语句。第 6 章讲述 C 语言的重要组成部分——函数,包括函数的定义、调用、返回、作用域等。第 7 章介绍指针和数组概念,包括指针和函数的关系、多维数组等。第 8 章和第 9 章讲述结构和表处理,包括联合、字段、线性链表、二叉树等。第 10 章介绍输入/输出和 UNIX 环境。

在编写方法上,每一章节开始有一内容概述,最后有一小结,并附有习题。课文中引用了大量例题,来解释和说明语法,并介绍程序设计方法。希望有助于读者很快地学会使用 C 语言。

由于编者水平所限,不当之处敬请广大读者批评指正。

编 者

一九九二年一月

目 录

第 1 章 初步介绍

1.1 什么是 C 语言	(1)
1.2 C 语言的特点	(2)
1.3 编程和运行	(4)
1.4 程序输出	(6)
1.5 语法规则	(9)
1.6 单词和词法	(11)
1.7 习题	(18)

第 2 章 C 的概述

2.1 变量、表达式和赋值	(20)
2.2 预处理语句 #define 和 #include	(23)
2.3 格式输入/输出函数 printf 和 scanf	(28)
2.4 控制流	(31)
2.5 字符输入/输出函数 getchar 和 putchar	(40)
2.6 函数	(46)
2.7 数组、字符串和指针	(51)
2.8 文件	(64)
2.9 实例	(69)
2.10 小结	(72)
2.11 习题	(74)

第 3 章 数据类型

3.1 基本数据类型	(77)
3.2 变量的说明	(78)
3.3 整数类型	(80)
3.4 浮点类型	(83)

3.5 字符类型	(86)
3.6 常量	(89)
3.7 小结	(93)
3.8 习题	(94)

第4章 运算符和表达式

4.1 表达式、语句和赋值	(97)
4.2 算术运算符	(100)
4.3 加 1 和减 1 运算符	(103)
4.4 赋值运算符	(106)
4.5 类型转换和强制运算符	(108)
4.6 求值顺序	(111)
4.7 关系运算符和表达式	(112)
4.8 逻辑运算符和表达式	(117)
4.9 条件运算符	(121)
4.10 按位运算符和表达式	(122)
4.11 小结	(129)
4.12 习题	(131)

第5章 控制流

5.1 复合语句	(136)
5.2 空语句	(137)
5.3 if 和 if_else 语句	(137)
5.4 while 语句	(143)
5.5 for 语句	(145)
5.6 实例	(147)
5.7 逗号运算符	(154)
5.8 do 语句	(155)
5.9 switch 语句	(157)
5.10 break 和 continue 语句	(159)
5.11 goto 语句	(160)
5.12 小结	(162)
5.13 习题	(163)

第6章 函数

6.1 函数定义	(173)
6.2 return 语句	(176)
6.3 函数的调用	(181)
6.4 作用域规则	(185)
6.5 存储类型	(188)
6.6 递归	(196)
6.7 预处理程序	(200)
6.8 小结	(208)
6.9 习题	(210)

第7章 指针和数组

7.1 指针	(218)
7.2 指针运算	(222)
7.3 一维数组	(224)
7.4 指针与数组	(227)
7.5 函数的指针参数	(228)
7.6 函数的数组参数	(231)
7.7 实例	(233)
7.8 字符串	(244)
7.9 多维数组	(251)
7.10 指针数组	(254)
7.11 指向函数的指针	(258)
7.12 命令行参数	(263)
7.13 不等长数组	(265)
7.14 小结	(267)
7.15 习题	(268)

第8章 结构、联合和枚举

8.1 typedef 的应用	(278)
8.2 结构	(281)
8.3 结构成员的存取	(284)
8.4 结构与函数	(287)

8.5 运算符总结.....	(291)
8.6 联合.....	(298)
8.7 字段.....	(301)
8.8 结构的初始化.....	(303)
8.9 枚举类型.....	(304)
8.10 实例	(313)
8.11 小结	(316)
8.12 习题	(318)

第9章 结构和表处理

9.1 引用自身的结构.....	(324)
9.2 线性链表.....	(326)
9.3 线性链表的操作.....	(328)
9.4 表处理函数.....	(332)
9.5 栈.....	(338)
9.6 二叉树.....	(344)
9.7 普通链表.....	(350)
9.8 小结.....	(355)
9.9 习题.....	(356)

第10章 输入/输出和 UNIX 环境

10.1 标准库和标准 I/O 文件	(363)
10.2 字符输入/输出	(364)
10.3 格式输出函数 printf()	(364)
10.4 格式输入函数 scanf()	(370)
10.5 函数 fprintf(), sprintf(), fscanf() 和 sscanf()	(373)
10.6 文件	(374)
10.7 前导文件 ctype.h	(381)
10.8 文件描述字和低级输入/输出	(387)
10.9 输入输出的改向和管道	(392)
10.10 C 编译程序和验证程序	(400)
10.11 C 的前景	(407)
10.12 小结.....	(408)

10.13 习题	(410)
附录 I	
ASCII 字符码	(414)
附录 II	
部分程序例题索引	(415)
附录 III	
参考文献	(418)

第 1 章 初步介绍

这一章介绍 C 语言的入门知识。首先简要介绍 C 语言的历史和演变过程,然后讨论 C 语言的五大特点,以便对 C 语言有一初步认识。为了便于学习和实践,这一章对编写和运行程序的步骤方法进行比较详细的解释,并对程序的输出给以说明和分析。这一章的后半部分介绍 C 语言语法规则,单词和词法,引入了标识符、关键字、运算符、字符串、注释、常量等概念。

1.1 什么是 C 语言

C 语言是一种通用程序设计语言。用 C 语言编写的程序易读性、正确性比较强,效率高,便于使用模块化结构,可移植性好。它既能用于科学计算,又能用于数据处理;既能用于编写系统软件,又能用于编写支撑软件和应用软件。C 语言与 UNIX 系统密切相关,UNIX 的操作系统、编译程序和应用程序大多由 C 语言写成,但作为一种语言而言,它独立于任一操作系统和机器。它可以用在各种不同应用领域编写计算机程序。

C 语言是由贝尔(Bell)实验室的丹尼斯·里奇(Dennic Ritchie)设计的,于 1972 年在 PDP—11 上实现。当时 C 语言是为 UNIX 系统设计的系统语言。C 语言的很多重要思想来源于 BCPL 语言。BCPL 是马丁·理查德(Martin Richards)于 1967 年提出的一种系统程序语言。肯·汤姆逊(Ken Thompson)基于 BCPL 设计了 B 语言,于 1970 年在 PDP—7 上写出了第一个 UNIX 系统。

BCPL 和 B 都是“无类型”语言。为了克服 B 的局限性,C 语言应运而生。C 引入了类型概念,扩充了一些功能,引进了更为全面

的控制结构等。在 C 语言中,基本的数据对象是字符、不同长度的整数和浮点数,并有由指针、数组、结构、联合和函数建立组合的导出数据类型。

C 语言提供了结构完善的程序所需的基本控制流结构,如复合语句,if-else 语句根据条件作出判定,switch 语句选择可能的情况之一,while 和 for 是先做测试的循环,do 是后做测试的循环,还有早期循环退出 break 等。

C 语言具有指针和地址运算的能力。使用指针是用 C 语言编程的一个重要手段。C 程序的基本组成是函数。函数能返回基本类型、结构、联合或指针的值。函数的参数是通过赋值来传递的,而且被调用的函数不会改变调用环境的实在参数。函数能递归调用,但函数调用不能嵌套。

C 语言本身仍在发展。计算机科学的进步对语言提出更加精确、更加现代的需求。语言标准化的工作推动 C 语言向一个“明确的、与机器无关的 C 语言定义”发展,同时又保留 C 的独特风格。

C 语言中 0 是自然起点。C 从 0 开始计数。C 用 0 代表“假”,用非 0 代表真。C 数组的下标以 0 为下界。C 字符串用 0 表示终止。C 指针用 0 表示空值。C 具有很多鲜明的特色。

1.2 C 语言的特点

C 语言是世界上七十年代以来最有影响的计算机语言之一,也是我国最常用的五种语言(COBOL、FORTRAN、PASCAL、C、ASSEMBLY)之一。C 语言得到越来越广泛的使用,归功于它的优点,特别是优良的“可移植性”。它不需要作大的修改就能在任何支持 C 语言的机器上运行。

概括起来,C 语言有如下特点:

1. C 语言简洁。C 是一种小型语言。在程序设计中,小则体现其美。C 所用的关键字,即被保留的字,比 PASCAL、ADA 少,然而

却被公认为比较强有力的语言。

另一方面,其运算符的表示方法也力求简炼,如加 1 运算符 ++ 使表达式紧凑,函数使用 {}, 而不用文字表示等,以至有时可以把多条语句浓缩成一条语句或者一个表达式。正如 C 设计者指出的,把 C 的规模压缩到较小会带来一些好处。由于对语言本身的描述也简单,所以易于学习,便于理解和使用,使程序员能够做到真正地规范地用上整个语言。软件生产率研究表明,程序员平均一天仅能够写出少量的语句。语言的简洁自然地提高了程序员的潜在生产率。

2. 类型丰富、结构完善。C 语言具有丰富的数据类型和控制流结构,C 不仅有字符型、不同长度的整型和浮点型数这些基本数据类型,还有随着指针、数组、结构、联合和函数组成的导出数据类型。C 提供了一套齐全的控制流结构,如 if、if .else、switch、while、for、do、break 等,以支持良好的程序结构。

3. 表达能力强。C 通常不完全被看作是一种高级语言,却被认为是相对“低级”的语言。这是指它可以操作由计算机才能处理的对象类型,如字符、数字和地址。对这些对象加以组合和修改可以完成通常由机器指令来实现的算术和逻辑运算。C 语言提供指针和地址运算能力,指针可作为数据来处理,所以体现出功能强,效率高。C 本身并不提供输入输出设施,这些高级机能由显式地调用标准库函数来实现。

4. 适用于模块化结构。C 支持一种程序结构——外部函数。函数是 C 程序的一种重要形式。C 语言程序都由同一种基本元素——函数组成。在源程序中函数之间数据传递方法,可以通过参数和返回数值来传递,也可以通过外部变量和静态变量来传递。函数能存在于分开的几个不同的源文件中,它们可以单独编译。任何函数都可以递归调用。但 C 不允许函数嵌套。C 通过使用在文件中的静态变量提供某些形式的有限保密。配合上 UNIX 环境,C 能方便

地支持用户定义的函数库和模块结构式程序设计。

5. 可移植性好。可移植性是指程序可以从一个硬件环境不加以修改或稍加修改就可移到另一种环境上运行。虽然 C 适合用在许多计算机机型上,它却独立于具体的计算机体系结构。灵活地运用预处理程序可以提高程序的可移植性。C 语言的编译程序简单而紧凑,而且不同机器上的编译程序的共同性达 80%,这就提高了语言的移植能力。现有的多数计算机都直接支持 C 语言的数据类型和控制结构,而为支持程序实现的运行库是很小的。

当然,象其它语言一样,C 也有缺点。有一部分语法不够合理,有一些运算符优先级不合适。某些运算符,例如 * 和 = 等具有多重用途。C 语言属于“弱类型”,类型转换比较随便,它允许编译程序在表达式和参数表中重新安排计算顺序。它不具有自动数组边界检查。在制订了标准以后,标准要求适当地说明和隐式转换,类型检查已得到加强。一些高质量的编译程序已对此提出要求。对不兼容的类型转换,编译程序会给出错警告。

1.3 编程和运行

什么是程序?程序就是计算机指令的有机序列,它规定了计算机解题的工作顺序。它是在计算机上解某个题目时所必需的一组完整的指令,亦即迫使计算机执行给定操作的一组控制命令。计算机程序可定义为适合计算机处理(包括计算或控制)的指令序列。程序设计是选用合适的算法和数据结构来编写程序的过程。在程序设计中,所有步骤都必须详细地规定清楚,计算机不容许含糊其词。计算机要求用一种可以理解的、精确的语言把这些命令写出来。程序反映了解题的具体步骤。程序设计从明确问题的技术规范开始,直到获得一个正确的程序为止。编码就是编写代码来实现设计思想的过程。

一般地说,程序设计可以分为以下几个步骤:

1. 明确问题的技术规范,
2. 找出解决该问题的算法,
3. 用计算机语言给这一算法编码,
4. 测试结果。

计算机是一部能够存储信息和执行命令的数字电子机器。从功能上分,机器由三个主要部分组成:内存、处理器和输入/输出外部设备。高速的内存能够存储信息和指令。处理器能够翻译从内存里取出的指令。输入/输出设备从机器外部的部件中提取信息,并向这些部件提供信息。典型的输入设备有终端键盘、磁盘驱动器和磁带驱动器。典型的输出设备有终端屏幕、打印机、磁盘驱动器和磁带驱动器。

驻留在机器内部的是一套专门的程序,称为**操作系统**(operating system)。我们在本书中假定自始至终采用 UNIX。操作系统好比交通警察一样,指挥着日常车辆行人的流通和行进。在 UNIX 环境下有很多软件包,包括 C 编译程序和各种各样文本编辑程序。主要的文本编辑程序称为 *vi*。编辑后得到的程序称为源程序。它要放在某个文件中,该文件称为源文件。文件名通常以 .C 结尾。UNIX 环境下 C 语言编译程序是通过 *cc* 命令来进行引用的。因为 *cc* 命令调用编译程序,所以该命令的名字也就是编译程序的名字。因而,“C 编译程序”和“*cc* 编译程序”可以互换使用。简洁地说,编译程序把源代码翻译成为可执行的目标代码。在该系统中得到的目标代码自动地放在可执行文件 *a.out* 中。运行 *a.out* 即可得到结果。

编写和运行 C 程序的步骤

1. 产生一个文本文件,文件名以 .c 结尾,将一个编辑好的 C 程序放入此文件。例如:

vi example.c

vi 是 UNIX 屏幕编辑程序。为了编写程序,读者应该选用一种编辑程序,学会使用插入和修改文本等编辑指令。

2. 使用 `cc` 编译程序对源程序进行编译。打入命令

`cc example.c`

如果在代码里没有错误,这时就会自动产生一个可执行的文件 `a.out`,程序就可以执行了。如果存在错误,整个过程再从第一步开始进行修正后重做。编辑过程中所出现的错误称为语法错误。

3. 运行程序。这是由下列命令来完成的:

`a.out`

如果程序编译正确,那么就开始执行程序。在一般情况下,如果程序里没有运行错误,程序就一直执行到完,系统提示会再现在屏幕上。如果由于某个原因需要修改程序,那么整个过程再从第一步做起。

程序中可能出现不同类型的错误。编译程序能够找出语法错误。其它错误,例如由于程序算法有错或给定的初值、参数不合适而造成的错误,则在执行程序的过程中才暴露出来。后者称为运行错误。例如,在一个程序的代码里出现了被 0 除,就会显示运行错。

1.4 程序输出

语言是交换信息、人机通讯的工具。

程序必须交流信息才有用处。本书第一个例题是在屏幕上打印出短语“from sea to shining C”(“从大海到闪光的 C”)。整个程序如下:

例题 1.1 程序 Shining_C。

```
/* * * 闪光的 C * * */  
main()  
{  
    printf("from sea to shining C\n");  
}
```

我们可以用屏幕编辑程序把这一程序送入到一个命名以 `.c`

结尾的文件中。文件名的选择应当有助于记忆。我们不妨取其名为 *sea.c*。打入命令

```
cc sea.c
```

其调用 *cc* 编译程序。如果代码没有错误,它就产生可执行的文件 *a.out*。现在,打入命令

```
a.out
```

即执行该程序,并且在屏幕上显示

```
from sea to shining C
```

经过编译的目标代码的缺省(default)文件名是 *a.out*。注意,如果这时你再编译下一个程序,*a.out* 会被重写过,产生新的可执行的代码。因此,如果你想保存在 *a.out* 里的可执行的代码,就必须取走(即重新命名)文件。最典型的做法是:如果源代码在文件 *sea.c* 中,那么就把在 *a.out* 里的相应的可执行代码移到 *sea* 中。具体来说,打入如下的命令:

```
mv a.out sea
```

上述编译程序有一简便方法。许多 UNIX 命令带有任选项(options),用一个减号作为标记。*cc* 命令具有很多任选项,其中之一是 *-o*。这一选择的作用是把目标文件放在紧跟在 *-o* 之后的文件中。因而,下面的命令

```
cc -o sea sea.c
```

将编译 *sea.c*,并且把可执行的代码存放文件 *sea* 中。最后,屏幕上显示出

```
from sea to shining C
```

程序 shining_C 的分析

```
main()
```

每一个程序都有一个名为 *main* 的函数,程序总是从 *main()* 开始执行。跟在 *main* 后面的圆括号()向编译程序表明这是一个函数。