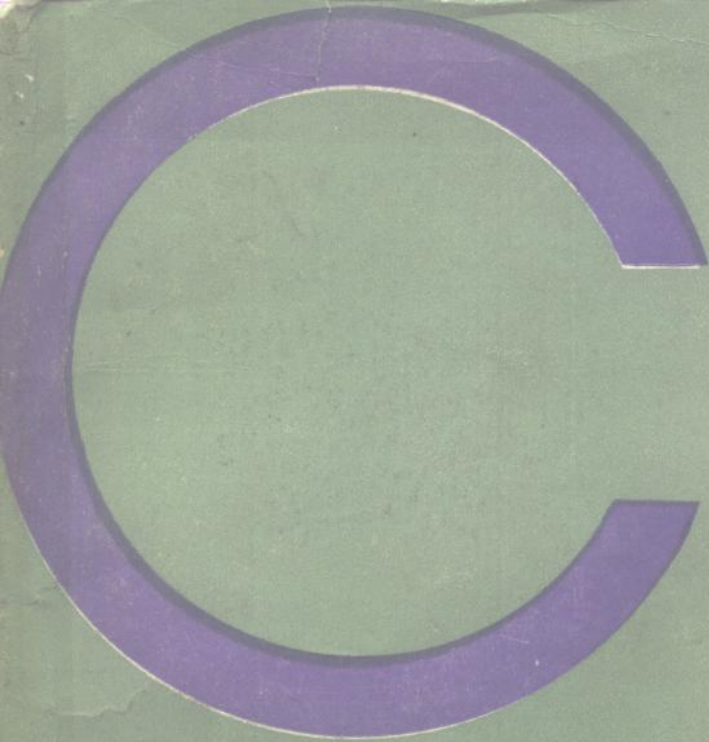




C语言

标准程序设计语言丛书

孙玉方 文强 编

中国铁道出版社

内 容 简 介

本书根据 ANSI C 标准草案,对 C 语言进行了系统而又严格的介绍。主要内容
包括: C 语言的变量、常量、类型及其转换;表达式与运算符;语句与控制流;函数;数
据存储类;数组和指针;结构、联合和枚举;预处理;输入/输出和文件处理;标准函数
库等。为了方便读者学习和掌握本书的内容,每章的最后都配有相应的练习题,以
便读者进行反思。为了使本书使用起来更为有力,我们在国内首次推出对书中例子
配有相应的软盘片,以便读者上机检验,加深感性认识。

本书可供计算机用户、大专院校师生及从事计算机工作的技术人员使用。

标准程序设计语言丛书

C 语言

孙玉方 文 强 编

中国铁道出版社出版、发行

责任编辑 郭 宇 封面设计 安 宏

各 地 新 华 书 店 经 售

中国铁道出版社印刷厂印

开本:850×1168 毫米 1/32 印张:13.25 字数:343 千

1989 年 8 月第 1 版 第 1 次印刷

印数:1—6,000 册 定价:5.70 元

前 言

正如大家都知道的,不论国内还是国外,只要使用计算机,就不可避免地要接触程序设计语言,因此程序设计语言的书籍就如雨后天春笋般地出版了。那么,我们又为什么要出版这套程序设计语言的丛书呢?或者说这套丛书又有什么特点呢?

在中国铁道出版社的大力支持下,全国计算机与信息处理标准化技术委员会程序设计语言分技术委员会——组织编写了这套“标准程序设计语言”丛书。丛书的编著者是多年来勤勤恳恳工作在程序设计语言标准化最前沿的人们,对来自对口国际组织——ISO/IECJTC1/SC22(国际标准化组织/计算机与信息处理技术委员会/语言分技术委员会)的各程序设计语言标准文本,我们都是国内的第一批接触者,并把它们及时地、严格地、具有立法权威地介绍给国内广大的软件工作人员。但我们在工作中深切地感到,能真正懂得文本真谛是一件很困难的事,标准文本对每个人的直觉是枯燥无味、晦涩难懂。那么在工作中,我们又怎能真正用好这些标准文本呢?

鉴于此,我们决心用通俗的语言编撰这套丛书。我们希望通过这套丛书能系统地并且严格地按照国际标准向读者介绍各有关程序设计语言,从名词、术语到内容上都严格地按照国际标准编写。全套丛书有统一的体例,介绍了有关该程序设计语言的发展历史及在我国如何进行标准化工作的。在内容上做到全面细致地讲解、介绍,可以学以致用,可作为教科书和手册使用,也可作为自学参考书。我们也把目前在国际化方面的最新动向介绍给读者,以使大家看到该语言的发展前景及方向。各册丛书的每一章都配有相应的练习题,使读者可以考验自己学到了多少,可以进行反思。我们同时也注意到在国家标准和国际标准中出现的一些难

以理解的“边角料”，并给予了解释，这样做有助于加深对标准的理解，有助于编译程序的开发。

在这里也想告诉读者一个好消息，在 1987 年召开的 ISO/IEC JICI/SC22 第二届年会上，通过我国代表团及各非英语国家代表团的努力，大会决定今后新推出的各项程序设计语言国际标准中都要考虑多字节字符集的加入问题。我们盼望着在国际标准中汉语将占有它应有的位置，盼望着这套丛书中将逐渐加入汉字字符的前景。

为使丛书使用起来更为有力，我们在国内首次推出对每本书中的例子都配有相应的软盘片，便于读者上机检验，加深感性认识，但考虑到经济负担，需要者可以直接和出版社联系购买。

我们希望这套丛书对我国广大的计算机用户、大专院校师生及从事计算机工作的技术人员都是适用的，希望成为你们的好助手。也希望大家在使用中把发现的不足之处及问题及时的告诉我们，以改进我们的工作。谢谢！

全国计算机与信息处理标准化技术委员会
程序设计语言分技术委员会主任委员

吴庆宝

1988 年 3 月 25 日于北京

作 者 的 话

随着 UNIX 操作系统的广泛流行及微型计算机的普及推广, C 语言作为 UNIX 操作系统的孪生兄弟, 正以新的姿态出现在程序设计语言及软件工具研究、开发和使用领域里。它的简洁、高效、可移植性等众多优点越来越受到软件工程技术人员的赞赏, 并得到越来越广泛的应用。今天它已是几种最常用的语言之一。

C 语言原是 AT&T (美国电报电话公司) 属下 Bell Labs. (贝尔实验室) 的 D·里奇 (Dennis Ritchie) 为开发 UNIX 操作系统而独立设计并实现的。随着它的广泛流行, 新的特点不断加入其中, 实现的版本也越来越多。鉴于 C 语言的重要性, 它本身存在的一些缺陷, 以及版本不统一等问题, ISO (国际标准化组织) 认为有必要制定有关 C 语言的国际标准。这一提议得到世界各国计算机界有关专家的赞同。在我国也于 1984 年 11 月在苏州召开了第一次有关 C 语言的专题学术会议, 在会议上我们和有关同志曾对 C 语言的标准化提出过一些建议并对此进行过讨论。从世界范围来讲, C 语言的标准化也有其困难的一面。这是因为 C 语言原有的文本不够完善和严格, 经过一些年又实现了许多版本。这样, 新的标准文本既要提出新的一致版本, 又要照顾到以前在不同版本上开发的大量 C 语言软件的生存问题。所以直到 1985 年 7 月 1 日才由美国提出了 C 语言标准化的一个非正式的建议草案。实际上它是一个有关 C 语言标准化的工作报告。

虽然这一非正式的文本草案需要作不少修改, 还要经过一段时间才有可能成为正式的文本, 但是其实质性的内容应该说是经得起考验的。这个非正式草案吸收了以前一些非正式文本, 特别是 D·里奇的文本中的合理成分, 并作了适量的扩充、修改。

为了向国内读者介绍标准 C 语言, 我们撰写了本书。全书共

分十一章，书后有四个附录。每章内容侧重于语言的一个或若干个方面，它的开头有本章概述，结尾处有本章的小结，提出一些重要之点，最后给出一些习题，供读者思考。具体讲

第一章引论，对 C 语言的演变史和特点作了概述，然后列举了几个具体的 C 语言程序，并借此引入了 C 语言的一些成分。为了使读者能够上机演习，还讲述了 C 语言的编译和运行方法。在本章的末尾逐渐引进 C 语言的一些基本词法。

第二章讲述 C 语言中变量的基本类型说明及各种基本常量，并介绍了各种类型转换方法。

第三章介绍了 C 语言中一些基本的运算符和相应的表达式。最后给出了所有运算符的优先级和结合性。

第四章讲述了 C 语言中所用的语句和控制结构。

第五章介绍 C 语言中重要成分——函数。从其定义到说明，从调用方式到返回类型都作了详细介绍。

第六章侧重于数据存储类，这是 C 语言与其它语言明显不同的地方。介绍了 C 语言中四种存储类，最后集中介绍了标识符的辖域、联接等概念。

第七章介绍 C 语言导引类型中重要的两种——数组和指针。这两种数据类型有着密切关系。本章对于指针作了十分详尽的介绍。

第八章介绍了三种类型：结构、联合、枚举以及它们的使用。

第九章侧重于 C 语言的预处理功能，包括文件蕴含、宏定义和宏替换、条件蕴含等。这也是 C 语言与其它语言的不同之点。

第十章输入/输出、文件处理与第十一章的标准函数库本来不是 C 语言编译的法定成分，而是由 C 语言所在的环境系统（如操作系统）提供的。但为编写一个完整的 C 语言程序，这些成分又是必不可少的。所以在 ANSI C 标准草案中对它们作了更为明确的规定。

在附录中对标准草案中提出的语法现象作了汇总，并给出了草案中与现有的实现重要的不同之点。最后列出了编写本书所用

到的参考文献。

由于我们所得到的仅仅是由美国提出的非正式的建议草案(在正文中为了简练我们暂称为 ANSI 标准草案),一些新的成分及限制还未见到具体的实现,所以给我们的编写工作带来不便。特别是其中有一些本来就是与具体实现有关的,而我们又不能一一列举具体实现的不同之处,更是增加了我们行文的困难。为此在涉及这些问题时往往要加一些解释,以免读者按草案文本中的约定编写程序,而在目前实现的版本上编译和运行,从而产生一些不必要的麻烦和误解。

本书的例子我们尽量经过现有计算机上的编译程序验算。其中许多例子本身就描述了一个完整的算法。

本书编写时具体参照了美国提出的非正式标准草案(ANSI X3J11 Standard Committee)。由于我们水平有限,对标准草案研究不够,理解也不一定正确,所以文中不当之处可能不少,恳请读者批评指正。我们也有志于根据读者的反映及正式标准文本对本书作全面的修订。

信息工程学院周锡龄院长及全国计算机与信息处理标准化技术委员会程序设计语言分技术委员会吴庆宝主任委员在百忙之中对本书作了严格的审核,提出了许多宝贵的建议和批评,谨致以由衷的感谢。

编 者

1988.1

目 录

第一章 引 论	1
第一节 C 语言的发展及特点.....	1
第二节 C 语言程序一览.....	3
第三节 C 语言程序编译和运行.....	12
第四节 标准 C 语言的基本词法及语法成分.....	15
第五节 语法规则的表示.....	19
第六节 小 结.....	20
练 习.....	21
第二章 变量、常量、类型及其转换	23
第一节 标识符.....	23
第二节 基本数据类型和变量说明.....	24
第三节 const 和 volatile.....	32
第四节 常量.....	35
第五节 字符串直接量.....	42
第六节 类型转换.....	44
第七节 小 结.....	49
练 习.....	50
第三章 表达式与运算符	52
第一节 表达式和简单赋值运算符.....	52
第二节 基本类型初始化.....	56
第三节 基本算术运算符.....	57
第四节 关系运算符与相等性比较运算符.....	61
第五节 移位运算符.....	62
第六节 按位运算符.....	63
第七节 逻辑运算符.....	65

第八节	增量运算符	66
第九节	条件运算符	69
第十节	复合赋值运算符	71
第十一节	运算符优先级与结合性	73
第十二节	副作用与序点	75
第十三节	小 结	77
练 习		78
第四章	语句与控制流	81
第一节	表达式语句	81
第二节	复合语句	82
第三节	选择语句之一——if 语句	83
第四节	迭代语句之一——while 语句	88
第五节	迭代语句之二——for 语句	91
第六节	迭代语句之三——do-while 语句	95
第七节	标号语句和转向语句	96
第八节	选择语句之二——switch 语句	101
第九节	小 结	104
练 习		105
第五章	函 数	108
第一节	函数定义	108
第二节	函数返回值的类型及其说明	114
第三节	函数调用与说明	123
第四节	函数模型	135
第五节	参数个数可变及前导文件<stdarg.h>	138
第六节	递归函数与函数的递归调用	144
第七节	C 语言的分程序结构与辖域	154
第八节	C 语言程序的部分编译	157
第九节	小 结	160
练 习		161
第六章	数据存储类	163

第一节	自动类和寄存器类	163
第二节	外部类	165
第三节	静态类	170
第四节	静态及外部类数据的初始化	174
第五节	类型定义	175
第六节	标识符	177
第七节	小 结	183
	练 习	183
第七章	数组和指针	186
第一节	数组的说明和使用	186
第二节	指针和地址	190
第三节	指针和数组	202
第四节	字符指针和字符串	210
第五节	构造复杂的数据类型	213
第六节	命令行参数	225
第七节	小 结	228
	练 习	229
第八章	结构、联合和枚举	231
第一节	结构的说明和使用	231
第二节	位域(bit field)和结构的存储映象	243
第三节	结构的自我引用	246
第四节	联合类型	255
第五节	不完整的结构或联合区分符	258
第六节	聚集数据结构的初始化	260
第七节	枚举类型	265
第八节	小 结	268
	练 习	269
第九章	预 处 理	271
第一节	简单宏定义和宏替换	271
第二节	带参数的宏定义	275

第三节	#include 预处理.....	281
第四节	条件蕴含.....	283
第五节	行控制.....	285
第六节	#error 预处理.....	286
第七节	#pragma 预处理.....	286
第八节	预处理程序提供的宏定义.....	286
第九节	小 结.....	287
练 习		287
第十章	输入/输出和文件处理	289
第一节	文件与数据流.....	289
第二节	基本文件操作.....	292
第三节	文件处理举例.....	304
第四节	格式输入/输出函数.....	313
第五节	小 结.....	323
练 习		323
第十一章	标准函数库	326
第一节	公共定义<stddef.h>.....	328
第二节	可变参数的宏处理<stdarg.h>.....	329
第三节	标准输入/输出函数库<stdio.h>.....	330
第四节	常用函数库<stdlib.h>.....	342
第五节	程序断言诊断<assert.h>.....	350
第六节	字符处理<ctype.h>.....	350
第七节	字符串处理<string.h>.....	353
第八节	中断信号处理<signal.h>.....	359
第九节	非局部转移<setjmp.h>.....	362
第十节	数学函数库<math.h>.....	365
第十一节	日期与时间<time.h>.....	369
第十二节	整型量大小的限制参数<limits.h>.....	375
第十三节	浮点类型的特征参数<float.h>.....	376
第十四节	地方特性<locale.h>.....	378

alloc.h bios.h conio.h dir.h . 9 .
dos.h graphics.h ioh mem.h

第十五节 小 结.....	379
练 习.....	380
附 录.....	381
附录 A C 语法汇总.....	381
附录 B 前导文件及库函数汇总.....	394
附录 C ANSI C 标准草案与现有 C 编译版本对照.....	403
附录 D 中英文术语对照表.....	405
参考文献.....	409

第一章 引 论

本章首先介绍 C 语言的发展史、特点,然后给出几个完整的 C 语言程序,以便读者对 C 语言程序建立一个整体概念。为了使读者能亲自动手编写并实现 C 语言程序,本章还简略地介绍了 C 程序的编译和运行方法。最后给出了 C 语言中最基本的一些词法和语法成分,并介绍了本书所用的语法规则表示方式。

第一节 C 语言的发展及特点

C 语言诞生于 1972 年。整个语言的设计及初版的实现几乎都是由 D·里奇一个人完成的,他当时正和 K·汤普逊(K. Thompson)一起开发和完善 UNIX 操作系统。

UNIX 操作系统原来主要是贝尔实验室内部的计算机专家为了他们编程序方便而开发的计算机软件系统。C 语言则是这个系统支持下的一个最基本也是最重要的工具。事实上,所有 UNIX 所提供的软件,包括 C 编译程序本身和 UNIX 操作系统都是用 C 语言实现的。

大约于 70 年代中期,UNIX 从贝尔实验室和 AT&T 内部走出来,开始在全世界流行。C 语言也就是因此流行并吸引了越来越多的软件开发人员。目前,C 语言在 AT&T 内外部都已成为一个“标准”语言。值得强调的是,C 语言之所以受到欢迎,并不是由于任何有计划的推销和宣传,而完全是由于使用者乐于接受它。到了 80 年代,在许多非 UNIX 的其它操作系统上也开始提供 C 语言。因此 C 语言的使用在某种程度上对目前的软件开发已产生了重大影响。

1983 年 C 语言与 UNIX 操作系统的开发者一起获得了该年度的 ACM 图灵奖,评审委员会对 C 语言的高度评价,确定了 C 语

言的学术地位。

为了便于与其它语言对照和比较,下面简单介绍 C 语言的演变历史。我们将发现,C 语言基本属于 ALGOL 语言体系。

ALGOL 语言是 1960 年提出的,它比 FORTRAN 语言虽然晚了几年,但由于其结构非常严密,因此对于后来的程序设计语言均有相当大的影响。但是由于其过于复杂和严密,在当时并未被人们广泛接受。1963 年 CPL (Combined Programming Language) 语言仿照 ALGOL 语言设计出来了,但其仍然过于庞大和复杂,难于学习和编写程序,所以亦未流行。1967 年提出了 BCPL (Basic CPL) 语言,它吸取了前两者的一些精华,但是又过于简略,缺乏通用性。K. 汤普逊为了用高级语言编写操作系统而基于 BCPL 语言于 1970 年设计了 B 语言。不过,由于 B 语言过于简单而且与 PDP11 的许多特性相违,所以 D. 里奇于 1971 年在 B 语言基础上开发了 C 语言。C 语言扩充了 BCPL 语言和 B 语言的一些功能,加进了一些基本数据类型和构造类型,使之更能直接而且清晰地表现客观世界所存在的各种对象。它也引进了更为全面的控制结构,从而为编写具有良好结构的实用程序奠定了基础。C 语言具有许多特点,使它能后来居上而为大家所喜爱。其优点主要有以下几个方面。

1. 简洁

C 语言很少提供直接处理复合对象,如作为整体来看的字符串,集合、表或者数组等的操作。C 语言本身也没有提供输入/输出手段,这些都由其运行的宿主环境(如 UNIX、DOS、VMS 等)提供。它也只提供单线的控制流构造,而不提供多任务、并发同步等操作。这样其编译程序就比较简洁。

2. 高效

C 语言是一种较“低级”的语言,它的一些操作直接对应于实际机器所执行的动作。在许多方面它具有汇编语言的特性,因而能充分反映机器硬件功能,所以代码效率极高。统计表明用 C 语言编写的程序编译后所得到的代码仅比用汇编语言直接编写的代

码效率低 15~20%。所以 C 语言足以替代汇编语言来编写最讲究效率的那部分软件,比如操作系统、编译程序等。最明显的例证是 UNIX 系统本身,它从内核到外层几乎全部都是用 C 语言写的。

3. 可移植

虽然 C 语言能与计算机的许多硬件能力相适应,但它却是独立于具体计算机体系结构的。因此,只要稍加小心便能书写出便于移植的程序。C 语言及 UNIX 系统的流行与这一特性密切相关。

C 语言还有其它一些优点,这里就不一一赘述了。C 语言和其它语言一样,也有它的缺点。例如,某些运算符优先顺序与习惯不完全一致;类型自动转换存在不稳定因素等等。尽管如此,对于多种应用来说,C 语言仍然是非常有效且非常有表达能力的语言。

第二节 C 语言程序一览

在以后各章中,我们试图把要介绍的内容排成一个合理的顺序,使每个章节尽可能地只依赖于前面的介绍。但这很困难,特别是如果要举出有意义的例子,必然要用到后面介绍的某些内容,比如输入/输出函数、数组、预处理语句等等。因此,在详细介绍 C 语言各种成分之前,先用一节简单介绍若干 C 语言程序例子和相应的成分(当然并不是 C 语言程序设计的速成介绍)。这样一方面可使读者对 C 程序有一个整体和概括的认识,以加深对局部成分与整体关系的理解。另一方面也可使我们在逐步深入并注重于某些主要成分介绍时不至于分散精力再去介绍一些其它的语法现象和语言成分。当然,这样做也可能使某此内容有些重复。希望这种重复对读者会有所帮助。下面就来看几个具体的 C 语言程序。

[例 1—1] 求三数之和。下面是其程序。

```
main ( )  
{  
  
    int a, b, c, sum;
```

```

a = 1000;
b = 100;
c = 10;
sum = a + b + c;
printf("The sum is %d\n", sum);
}

```

下面对本程序作一解释：

main()

C 语言中一个程序可以分成若干个文件（在 UNIX 操作系统中，这些文件名通常以 .c 结尾，作为 C 语言源程序之标志）。每一个文件中可以包含若干个函数。而在组成整个程序众多的函数（本例中程序只有一个文件，而这个文件中只有一个函数）中有且只能有一个函数的名字为 main。程序从此函数开始执行。main 后的一对圆括号，向编译程序表明这是函数。

{

这里开花括号，它等同于 PASCAL 语言及其它语言中的 begin，与闭花括号（等同于其它语言中的 end）一起把一组说明和语句（称为复合语句）括在一起。

```
int a,b,c,sum;
```

这是说明语句，说明 a、b、c 和 sum 是四个整数类型(int)变量。说明语句最后以“;”结尾。

```

a=1000;
b=100;
c=10;
sum=a+b+c;

```

这是四个赋值语句。在 C 语言中赋值号为“=”。赋值语句（详见第三章第一节）是 C 中的基本语句。在每一个语句的后面都带有“;”。注意，在 C 语言中，分号是语句终止符，是语句的一部分，而不像某些别的语言中仅作为分隔符来使用。所以，分号在 C 中是不可省略的。

```
printf("...");
```

这是一个打印语句。printf() 是一种子程序（函数）。与 C

语言一起，系统维持了一个函数库，其中有不少输入/输出函数。`printf()`便是其中的一个函数，它用来进行格式输出。本例主要是通过 `printf()` 把 `sum` 中的内容按十进制整数(`%d`)形式输出。这里 `%d` 是输出格式标志。`%d` 前的字符串原样输出，所以本例的输出是：`The sum is 1110`。在双引号括起的字符串中，`\n` 是换行字符。

`printf()` 是最常用的一种输出函数，下面列出它的最基本和最常用的格式(见表 1—1)，以便参考(详见第十章第四节)。

`printf()` 的一些主要打印格式 表 1—1

格式符	对应之参数按何种方式印出
<code>%c</code>	作为字符
<code>%d</code>	作为十进制整数
<code>%e</code>	作为浮点数(按数学表示形式, 如 <code>2.03e+2</code>)
<code>%f</code>	作为浮点数(按普通表示形式, 如 <code>203.00</code>)
<code>%s</code>	作为字符串
<code>%%</code>	<code>%</code> 字符本身

要注意的是，一般在 `printf()` 中若出现几个格式符，就要有几个对应之参数与之配对。

}

这是函数体的终止标志。

〔例 1—2〕 把小写字母转换成大写字母。

在大多数 C 语言系统中，字符的取值为相应 ASCII 码的值，比如字母 `a` 有编码值 97。如果把字符编码值看成是整数，则可以让它参加一般的算术运算。在 C 语言中，字符的编码值用字符本身外加单引号表示，比如在采用 ASCII 码对字符编码的系统中

`'a' = 97`

`'z' = 122`

⋮

由于 ASCII 编码表中大写字母和小写字母都是顺序编排的，所以 `'Z' - 'z'` 与 `'A' - 'a'` 有同样的值。利用这些特性可以很容易把大写字母转换成对应的小写字母。下面是其程序。