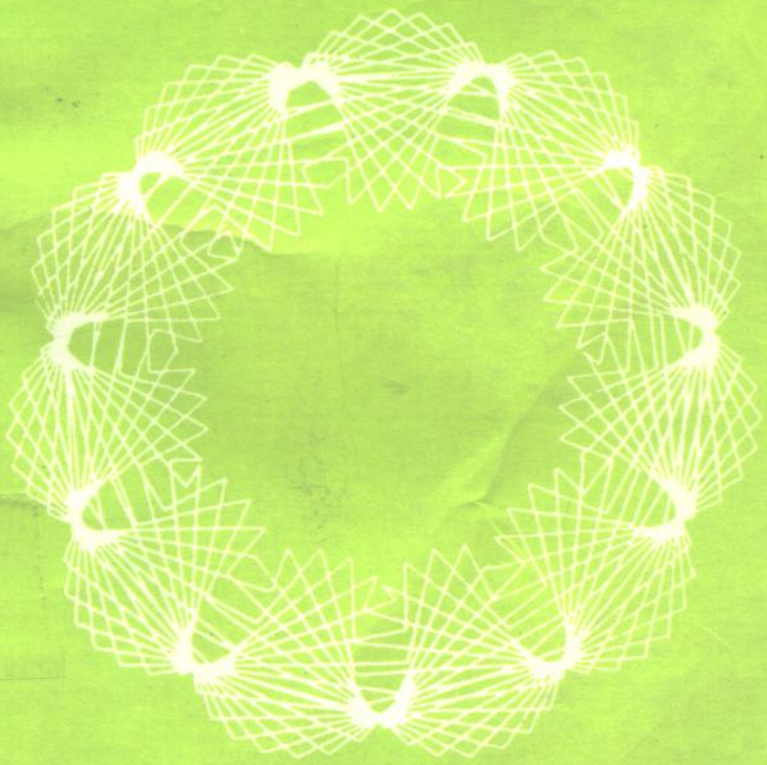
计算机应用基础系列丛书

C语言

程序设计基础

曹兰斌　谭月辉　王希武　郑利强　编



電子工業出版社

计算机应用基础系列丛书

C语言程序设计基础

曹兰斌　谭月辉　王希武　郑利强　编

电子工业出版社

(京)新登字 055 号

内 容 简 介

本书系统地介绍了 C 语言的数据类型、语句和程序设计方法，深入浅出、循序渐进、示例丰富、通俗易懂。

全书共分九章。第一章概述了 C 语言的特点、程序结构和上机步骤，第二章介绍了数据类型、运算符和表达式，第三章阐述了语句及基本程序设计方法，第四章阐述函数，第五章阐述数组和指针，第六章阐述结构和联合，第七章介绍文件，第八章介绍编译预处理，第九章介绍 Turbo C、Microsoft C，最后给出了包括 C 库函数在内的五个附录，供读者查阅。

本书可作为计算机应用专业和计算机 C 语言培训班的教材，也可作为计算机应用科技人员自学、使用的参考工具书。

计算机应用基础系列丛书

C 语言程序设计基础

曹兰斌　谭月辉　王希武　郑利强　编

责任编辑　王昌铭

*

电子工业出版社出版(北京万寿路)

电子工业出版社发行　各地新华书店经销

河北省望都县印刷厂印刷

*

开本：850×1168 毫米 1/32　印张：9　字数：255 千字

1995 年 8 月第 1 版　1995 年 8 月第 1 次印刷

印数：7000 册　定价：11.00 元

ISBN 7-5053-3012-8/TP.1045

前　言

随着计算机技术的迅速发展，计算机的应用范围日益广泛。为适应计算机应用工作的普及和发展的需要，我们编写了本丛书。本丛书共十二册，涉及到计算机应用基础知识的各个方面，内容丰富，实用性强，是全面学习和掌握计算机应用知识难得的普及性读物。其内容包括：《操作系统及应用》、《微型机常用通用软件》、《计算机常用汉字输入方法》、《Pascal 语言程序设计基础》、《计算机硬件基础》、《计算机软件基础》、《C 语言程序设计基础》、《dBASE 数据库应用基础》、《ORACLE 数据库应用基础》、《计算机网络应用基础》、《微型机常见故障维修实例》、《微机屏幕提示信息英汉译义》。

C 语言自 70 年代初问世以来，发展相当迅速，相继出现了诸如 ANSI C、Turbo C、Microsoft C、C＋＋、Visual C＋＋等若干实用版本，以简洁、紧凑、灵活、使用方便，表达能力强，可移植性好为突出特点，引起了技术人员的关注。C 语言常常被称为中级语言，它既有高级语言特点，又有汇编语言特点，把高级语言的基本结构与低级语言的机器性能描述成功地结合起来，深受软件开发者欢迎，近年来得到了广泛的应用，表现出很强的生命力。因此，我们编写了此书，供读者使用，为普及和提高应用计算机打下一个基础。在编写本书时，我们本着通俗易懂、深入浅出的原则，力求简明、实用、示例丰富，且全部举例的程序用 Turbo C 上机通过。本书既可作为计算机应用专业和计算机培训班的教材，又可作为各层次人员自学和参考的工具书。

本书由曹兰斌、谭月辉、王希武、郑利强合编，全书由曹兰斌教授统稿，由张喜林，刘世诚，王大刚主审。

由于编者水平有限，望读者不吝赐教。

编者

1995 年 2 月

目　录

第一章 绪 论

这一章我们介绍C语言的发展历程、特点，从简单的C语言程序对C语言作一概括了解，最后介绍C语言的上机步骤。

1.1 C语言的发展和特点

1.1.1 C语言的发展历程

提起C语言，就不能不提到UNIX系统。因为从某种角度可以说，没有UNIX系统就没有C语言的产生和发展；没有C语言也没有UNIX系统的今天。它们之间有着相互依存、休戚与共的紧密关系。

UNIX系统是美国贝尔实验室的K·Thompson和D·M·Ritchie从1969年开始，用不到两个人年的时间研制成功的。第一版本是在GE635机上产生并通过纸带把可执行代码传送到DEC的PDP-7机上的，在UNIX上实现了汇编语言后，UNIX系统又以汇编语言编写。由于汇编语言的不可移植、描述问题的效率低、可读性差等缺点，K·Thompson决定开发一种专门语言来描述UNIX系统。但一般高级语言难以实现汇编语言的某些功能，如内存寻址、位操作等。而这些又是设计系统软件所必需的。于是，既具有高级语言特性，又具有低级语言特点的B语言在1970年在PDP-11/20上由K·Thompson实现了，并用B语言写了UNIX操作系统和绝大多数实用程序。B语言主要思想源于BCPL(Basic Combined Programming Language)语言，BCPL是M·Richards在1968年对CPL(Combined Programming Language)语言进行简化后发表的一种单一

数据型语言，今天仍在使用(主要是欧洲)，但是有下述几个主要缺点：

1. 大多数机器是字节编址的，而 B 却面向字，这样就妨害了对单个字节的存取。

2. 现代高级语言要求有强有力的类型结构，而 B 语言的无类型性(或者说只有一种机器字类型)，在描述客观世界许多事物时遇到相当多的困难。

3. B 编译程序产生的是解释执行的代码，运行速度较慢。

还有一些次要的原因使 B 语言没有盛行起来，从而导致 D·M·Ritchie 于 1971 年开始开发 C 语言，并在 1972 年 C 语言投入使用。1973 年 K·Thompson 和 D·M·Ritchie 把系统用语言重写了一遍。

后来，C 语言多次作了改进，但主要还是在贝尔实验室内部使用，直到 1975 年 UNIX 第六版公布后，C 语言的突出优点才引起人们的普遍注意。1977 年出现了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》，使 UNIX 系统迅速在各种机器上实现，C 语言也得到了广泛推广。以 1978 年发表的第七版 UNIX 系统中的 C 编译程序为基础，B·W·Kernighan 和 D·M·Ritchie 合著了影响深远的名著《The C Programming Language》。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础，被称标准 C。1983 年，美国国家标准化协会(ANSI)根据 C 语言问世以来各种版本对 C 的发展和扩充，制定了新的标准，称为 ANSI C。ANSI C 比原来的标准 C 有了很大发展。1988 年 B·W·Kernighan 和 D·M·Ritchie 修改了他们的经典著作《The C Programming Language》，按照 ANSI 标准重新写了该书。现在流行的各种 C 语言版本都是 ANSI C 的实现和扩充。

虽然最初的 C 语言是附属于 UNIX 系统且在 PDP-11 上实现的，但目前的 C 语言却独立于 UNIX 系统，独立于 PDP-11 机而蓬勃发展，它适应的机种从 8 位微型机(如以 Z80 为 CPU 的 Cromem-

ce)直到巨型机(如 Cray-Ⅱ)。它附着的操作系统从 8 位微机上的单用户 CP/M 到大型机的 IBM VS/370。它与 FORTRAN、Pascal 等语言一样已经成了各种机器上共同使用的通用语言。

1.1.2 C 语言的特点

一种语言之所以能存在和发展,并具有生命力,总是有不同于其他语言的特点。C 语言也如此,它的特点是多方面的,人们从不同的角度可总结出众多特点,从全面考虑可归纳为:

1. 语言表达能力强

C 语言常被称为中级语言,这并不意味着它功能差,难以使用,或不如 BASIC、Pascal 等高级语言那样完善,也不是说它类似于汇编语言,或涉及到使用汇编语言时所遇到的有关问题。C 语言定义为中级语言,只是意味着它把高级语言的基本结构与低级语言的实用性两者结合起来了。

C 语言是面向结构程序设计的语言,具有结构化的控制语句,用函数作为程序模块以实现程序的模块化,符合现代编程风格.同时 C 语言允许直接访问物理地址、能进行位操作,能实现汇编语言的大部分功能,可以完成通常由硬件实现的算术、逻辑运算,反映了当前计算机的性能.因此,它是成功的系统描述语言和通用的程序设计语言。

2. C 语言简洁、紧凑,使用方便、灵活,易于学习和应用

C 语言是小而精的语言,只有三十二个关键字(见表 1-1),它们构成了 C 语言的全部指令。C 语言把一般语言的许多成分都通过显式调用库函数来完成。比如它没有 I/O 设施,也没有并行操作、同步或协同程序等复杂控制,而是提供了大量而有效的库函数。库函数可根据需要方便地扩充。C 本身运行时所需支持少,存储空间占得也少。

3. 数据类型丰富,具有现代语言的各种数据结构

C 语言的基本数据类型有整型(int)、浮点型(float)、字符型

(char)。在此基础上按层次可产生各种构造类型，如数组、指针、结构、联合等。用这些数据类型可以实现复杂的数据结构，如链表、树等。

4. C 语言生成的代码质量高

在代码质量上，C 语言可与汇编语言媲美，针对同一问题，用 C 语言编写的程序，其代码效率仅比用汇编语言写的代码低 10%～20%。由此 C 语言的程序运行效率很高。

5. 可移植性好

目前 C 语言在许多机器上实现，大部分都是由 C 语言编译移植得到的，不同机器上的编译程序大约 80%的代码是公共的。C 编译程序的可移植性，也就使 C 语言程序便于移植。正如荣获 83 年图灵奖时评论中所说的，"对于 UNIX 系统可移植性的关键贡献是 Ritchie 开发的 C 程序设计语言"。

C 语言的优点很多，但也有一些不足。如运算符优先级太多，不便于记忆，有些还与常规的约定有所不同；语法限制不太严格，类型检验太弱，不同类型数据转换比较随便，如整型量与字符型量及逻辑型数据可以通用，因此不太安全。然而限制严格，就失去了灵活性，这就要求使用 C 语言的人，对程序设计的方法和技巧更熟练一些，以保证自己的程序的正确性。

从应用角度来讲，C 语言不如 BASIC 容易掌握；不如 Fortran 或 Pascal 语言那么规范。PL/I 科学计算能力强；在商业和管理等数据处理领域不如 COBOL 语因为 C 语言的特长不在这些领域，对操作系统和系统实用程序及需要对硬件进行操作的场合，C 语言明显地优于其他高级语言，有些大型应用软件(如数据库软件)也用 C 编写。

从教学角度看，目前 C 语言在理论研究方面不及 Pascal、Algol 60 等语言，在"数据结构"和"编译技术"等课程中一般用 Pascal 和 Algol 语言举例。但是 C 语言也是理想的结构化语言，描述能力强，

而且"操作系统"多结合 UNIX 系统讲解，UNIX 与 C 语言的不可分，使 C 语言有可能取代 Pascal 和 Algol 60 成为被广泛应用的教学语言，且 C 语言还有更广泛的应用领域，因此更有生命力。

总之，尽管 C 语言有这样那样的不足，但仍不失为一个实用的程序设计语言，由于它突出的优点，而吸引人们对它倾注越来越多的关心。以致在国内外使用、研究 C 语言的人正迅猛增加，优秀的 C 语言版本和配套工具软件不断涌现。

1.2 C 语言程序设计格式和结构特点

1.2.1 字汇表和语法图

任何一种语言，都需要有自己的基本字汇表，C 语言也不例外。C 语言的基本字汇表由以下几部分组成：

数字：0、1、2、……、9

字母：A ～ Z、a ～ z

下划线：_

运算符：+、-、*、/、%、=、<、>、<=、>=、!=、==、<<、>>、&、|、&&、||、^、~、(、)、[、]、->、.、,、!、?、:、

关键字：ANSI 推荐了 32 个关键字(见表 1-1)。

表 1-1 ANSI C 的关键字

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

对于关键字，各种 C 语言版本都有自己的扩充.关键字都有固定的含义，不能作为一般标识符使用。

如同自然语言和其它高级语言一样，C 语言也有它特定的语法规则，利用基本字汇表中的符号和关键字，按照给定的语法规则，可以构造其它符号、语句和程序，编写出的程序是否合法，完全取决于它们是否遵守给定的语法规则。通常，表示语法规则的形式有两种。

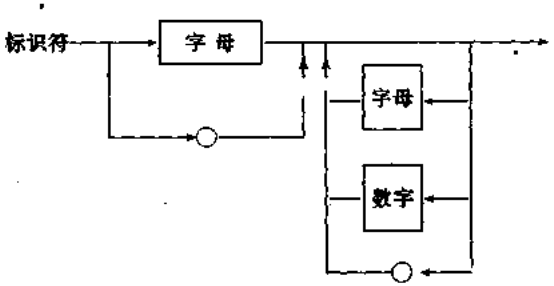
1. BNF(Buckus-Naur Form) 巴科斯范式

BNF 首先用于描述 Algol 60 的语法规则，沿用至今已有 20 多年的历史，它在一些语言的语法描述、这种方法的最大优点是清晰、严谨，在语言形式化描述，编译程序自动生成等研究领域中是一种最为有效的工具。

2. 语法图

语法图最早盛行于描述 Pascal 的语法规则，至今已有 10 多年的历史，这种方法的最大长处是直观形象，容易理解和掌握，在语言教学中是一个得力的工具，本书从讲述 C 语言角度出发，采用了语法图形式。

例如，在 C 语言中构造一个标识符的语法图：



按照语法图，26 个大小写英文字母、下划线及它们和数字的任意组合（第一个字符必须为字母或下划线）是合法的标识符，因此，a，_A，aBc，x12，Y_3 是合法的标识符，而 123，3_ab，#abc，! 45，a * bc 等是非法的标识符。

从上例可以看出，语法图是如何描述相应的语法规则的。在语法

图中，从箭头的入口到出口，按箭头的指向有一条或多条路线，按每条路线所定义的符号、语句和程序都是合法的。在开始学习使用C语言时，经常参照语法图，检查自己所编写的程序是否符合语法规则，可以使自己较快地掌握C语言的使用方法。

1.2.2 C语言程序结构

1. C语言程序的书写形式

计算机程序主要由两部分组成，即：

程序＝算法＋数据结构

其中算法是关于程序所要实现的目的的描述，它一般由一系列语句组成。这些语句可能根据描述的对象的要求以及语言本身的规则，构成复合语句、分程序、函数和过程等。数据结构是关于算法所要操作的对象(即数据)的描述，程序中出现的数据，或者是常量或者是变量，对数据的描述是对程序中用到的常量和变量进行的说明。

C语言是函数型语言，函数是构成C语言程序的基本单位。下面我们通过一个例子来分析C语言的程序结构。

[例1.1]

```
int count;              /* 定义变量 */
main()                  /* 主函数 */
{
  int a,b,sum;          /* 定义三个变量 */
  a=25; b=75;           /* 为a、b赋值 */
  sum=add(a,b);         /* 调用函数add,将得到的值
                           赋给变量sum */
  mul(a,b);             /* 调用函数mul */
  printf("sum=a+b=%d\tcount=a*b=%d\n",\
        sum,count);
}
int add(x,y)            /* 定义add函数 */
int x,y;                /* 定义形式参数 */
{
  int z;
```

```
    z=x+y;
    return (z);          /* 返回 z 的值 */
}
mul(int x,int y)         /* 定义 mul 函数 */
{
    count=x*y;
}
```

该程序的目的是显示两个整数 a、b 的和与积。

该程序执行输出：sum=a+b=100　　count=a*b=1875

从程序我们看出：

(1) 程序是由函数组成的，函数不能嵌套。

C 语言程序由若干个函数组成，组成程序的若干函数中必须有且只能有一个名为 main 的函数，各函数的位置无关紧要。C 语言程序总是从 main( )函数开始执行，通常我们总是把 main()函数放在程序中其它函数的前面。从技术上讲，main 一词并不是 C 语言的组成部分，但还是应当把它当作关键词来对待。

该程序包含函数 main、add 和 mul。

(2) 函数名后必须有一对圆括号"("和")"，这是函数标志。

(3) 函数必须由"{"开始，由"}"结束。

组成函数的语句必须由一对大括号括"{……}"起来。一个函数至少有一对大括号。

(4) 程序中的每个语句后必须有一个分号";"。

(5) C 程序书写格式自由。

一行内可写几个语句；一个语句也可以写在多行上，用"\" 作续行符，如程序的第 5 行和第 9 行。

(6) 可以在 C 程序的任何部分加注释，以提高程序的可读性。

注释使程序变得清晰，能帮助我们阅读和理解程序。给程序加注释是一个良好的编程习惯。C 语言注释由"/*"开始，至"*/"结束。注释可为若干行，但不允许嵌套。

程序的第 6、8、9 行是函数调用语句。其中 add()、mul()是用户