



IBM PC C语言简明教程

李文兵 年璋潮 编著

天津科学技术出版社

责任编辑：刘万年

IBM PC

C语言简明教程

李文兵 年海潮 编著

*

天津科学技术出版社出版

天津市赤峰道130号

天津武清永兴印刷厂印刷

新华书店天津发行所发行

*

开本850×1168毫米 1/32 印张11 字数277 000

1988年4月第1版

1988年4月第1次印刷

印数：1—12 800

ISBN 7-5308-0262-3/TP·8 定价：3.15元

内 容 简 介

C语言既是成功的系统描述语言，又是通用的程序设计语言。

本书分10章介绍了C语言的基本概念、语法规则和使用方法，并运用213个经过在IBM PC及其兼容机上PC DOS操作系统下实际调试的练习程序进行了说明。两个附录分别给出了行编辑的用法和出错信息。

本书在内容组织上注重系统性；在写法上注重简明易懂，在练习举例上注重可行性。

本书配有上机软盘，并由中国软件行业高档微机协会制作了教学录像片。本书适于作为大专院校、电大、计算机专业自学考试、计算机中专和有关培训班的教材；也可作为有关工程技术人员和管理人员的参考书或自学用书。

JS341/01

前 言

C语言是70年代Bell实验室为描述UNIX操作系统和C编译程序，而开发的系统描述语言。1975年UNIX第6版公诸于世后，C语言便被举世瞩目。由于C语言有：移植性好、应用性广、程序设计自由度大、表示方法简洁、软件工具丰富等优点，因此特别受到软件工程技术人员的喜爱，许多有名气的操作系统都是用它编写的。

目前C语言已被移植到小至8位机、大至巨型机，特别是8至32位微型计算机上，并能在各种操作系统下运行。

在我国，开设C语言这门课的大中专院校越来越多。为适应这种形势，又鉴于我国IBM PC及其兼容机用户较多的现状，我们编写了这本《C语言简明教程》。

在这本书的写作过程中，我们得到了天津师范大学计算机科学系王勤民副教授、中国软件行业高档微机协会徐国平高级工程师、中国科学院董洪皋高级工程师的鼓励 and 大力支持，在此，向他们表示衷心的感谢。

本书第八章由天津纺织工学院自动化系尤一鸣同志编写。贾秀荣同志负责全书的誊写与制图。

由于时间仓促、水平有限，错误和不妥之处，诚恳希望广大读者和有关专家指教。

作 者

1987.10 于天津

目 录

1 章 绪论	(1)
§ 1.1 C 语言	(1)
§ 1.2 C 程序	(6)
§ 1.3 上机操作	(12)
2 章 数据类型和简单程序设计	(19)
§ 2.1 整型数据	(19)
§ 2.2 字符型数据	(25)
§ 2.3 浮点数	(33)
§ 2.4 类型的混合运算	(40)
§ 2.5 数组	(43)
3 章 输入输出和库函数	(51)
§ 3.1 字符输入输出函数	(51)
§ 3.2 按格式输出函数	(57)
§ 3.3 按格式输入函数	(63)
§ 3.4 字符串处理函数	(70)
§ 3.5 文件处理函数	(76)
4 章 运算符和表达式	(81)
§ 4.1 表达式	(81)
§ 4.2 算术运算符	(82)
§ 4.3 位逻辑运算符	(88)
§ 4.4 赋值运算符	(92)

§ 4.5	关系运算符和等值运算符.....	(95)
§ 4.6	递增/递减运算符	(97)
§ 4.7	指针运算符.....	(99)
§ 4.8	逻辑运算符.....	(102)
§ 4.9	其它运算符.....	(104)
§ 4.10	运算符的优先级和结合性.....	(109)
5 章	控制语句	(112)
§ 5.1	if语句	(112)
§ 5.2	while语句	(119)
§ 5.3	for语句	(124)
§ 5.4	do-while语句	(131)
§ 5.5	break语句	(137)
§ 5.6	continue语句.....	(140)
§ 5.7	switch语句	(142)
§ 5.8	goto语句	(147)
6 章	宏预处理	(149)
§ 6.1	宏定义.....	(149)
§ 6.2	带参数的宏.....	(154)
§ 6.3	包含文件.....	(159)
§ 6.4	条件编译.....	(163)
7 章	函数调用	(170)
§ 7.1	函数值和return语句.....	(170)
§ 7.2	函数调用形式.....	(172)
§ 7.3	递归调用.....	(182)
§ 7.4	外部调用.....	(185)

8 章 变量的存贮类别	(193)
§ 8.1 自动存贮变量.....	(193)
§ 8.2 外部存贮变量.....	(199)
§ 8.3 静态存贮变量.....	(206)
§ 8.4 寄存器存贮变量.....	(211)
9 章 指针的用法	(216)
§ 9.1 数组的指针.....	(216)
§ 9.2 指针的运算.....	(222)
§ 9.3 指针参数.....	(226)
§ 9.4 指针数组.....	(236)
§ 9.5 函数指针.....	(241)
10 章 结构数据	(247)
§ 10.1 结构定义及其变量的初始化	(247)
§ 10.2 结构数组	(257)
§ 10.3 结构指针	(262)
§ 10.4 引用自身的结构	(269)
§ 10.5 字段结构	(275)
§ 10.6 联合	(277)
附录A 行编辑的用法	(284)
§ A.1 建立新的源文件	(284)
§ A.2 编辑已有源文件	(286)
§ A.3 行编辑命令	(287)
§ A.4 编辑键	(315)

附录B 出错信息	(320)
----------------	---------

§ B.1 设备信息	(320)
------------------	---------

§ B.2 命令或文件信息	(321)
---------------------	---------

§ B.3 编译信息	(322)
------------------	---------

§ B.4 链接信息	(339)
------------------	---------

§ B.5 存储空间信息	(339)
--------------------	---------

参考文献	(341)
------------	---------

1章

绪 论

§ 1.1 C 语 言

1. C语言的历史背景

C语言是70年代贝尔(Bell)实验室为描述UNIX操作系统和C编译程序,而开发的一种系统描述语言。

1969年,美国贝尔实验室的两个研究人员Ken Thompson和Dennis M.Ritchie,开始研究UNIX,用了不到两个人年的时间就研制成功了。

UNIX的早期版本是用汇编语言编写的。

1970年, Ken Thompson为了提高UNIX的可读性和可移植性,在BCPL语言的基础上,开发了一种新的语言,起名叫“B”。之所以叫这个名字,据说其中一个因素就是根据BCPL的字头。由于B语言存在着一些缺点,致使它并未流行起来。

从1971年开始, Dennis M.Ritchie用了一年左右的时间,在B的基础上开发了第一个C编译程序,1972年开始使用。因为这个编译程序是在B语言的基础上开发的,又因无论是在英文字母序列中也好,还是在BCPL这个名字中也好,排在B后面的均为C,故起名为“C”语言。

1973年, Ken Thompson和Dennis M.Ritchie两个人合作,把UNIX全部用C语言又写了一遍,这就为UNIX的移植和发展奠定了基础。之后, C语言虽经过多次改进,但主要还是在贝尔实验室内部使用。直到1975年UNIX的第6版本公诸于世后, C语言才被举世瞩目。

在这个时期,没有官方任何形式的宣传, C语言凭着它本身

的优点，就得到了广泛的应用。越来越多的程序编制人员放弃了已运用自如的FORTRAN语言、PL/1语言、甚至刚刚掌握的PASCAL语言或是APL语言，而偏爱上了C语言。

当时的C语言，同现在的标准版本，无论在功能上，还是在表示方法上，都有所区别。所谓标准版本，一般是指由Brian W.Kernighan和Dennis M.Ritchie著的《The c programming language》。这本书是基于1978年所发表的UNIX第7版本中的C编译程序而写的。这本书里所介绍的C语言，直到现在，大体上并没有什么变化；只是C编译程序的功能增加了两点：第一点是作为赋值语句和函数的参数使用了结构；第二点是增加了枚举类型enum，这是类似于PASCAL的标量scalar的类型。

虽然C语言开始是附属于UNIX的，是在小型机PDP-11上实现的，但到1978年以后，C语言便被移植到小至8位微机、大至巨型机Cray 1等大多数计算机上。目前，C语言已独立于UNIX和PDP-11，运行在各种操作系统上和各种机种上。特别是在8位至32位微机上使用非常普遍。

2. C语言的特点

(1) 移植性好 C语言作为一种标准语言之所以得到广泛应用，原因就在于C语言能被移植到多种操作系统上。

C语言之所以移植性好，不仅在于该语言适应范围广，还在于其处理系统很紧凑。这是因为作为语言，它没有定义输入输出，而且关键字用得少。

在C语言里，语法上仅仅定义了最基本的控制语句，而输入、输出和字符处理等基本函数并没有定义，它们是作为外部函数，链接到所编写的程序上才使用的。这一点，也是C语言处理系统较小、移植性高的原因之一。结果便使得目前C编译程序已在8位微机至Cray 1机等硬件上运行，并能在CP/M至MVS等操作系统上使用。

虽然C语言能够使用的环境是多种多样的,但是用C语言所编写的程序在这些环境之间进行移植时,还需要注意两个问题:

①各变量类型及其字长问题。各变量类型的字长因所用的CPU的字长而异,故移植程序时必须注意。

②外部函数的问题。作为外部函数的程序间的关系;在UNIX操作系统中如图1.1所示。图中的黑粗线是对于C程序员的UNIX,即操作系统的C程序的接口。这里分为sys、gen、stdio三个组。如果接口中函数的集合同UNIX完全相同,那么UNIX上所有应用程序将是 可以移植的。

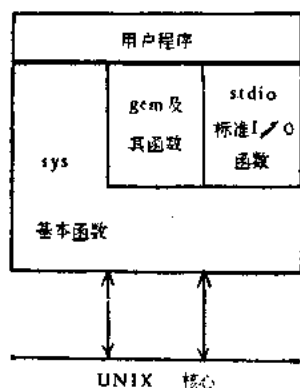


图1.1 外部函数程序间的关系

(2)应用性广 C语言应用性很广,用C语言编写的源程序中最大且最有名气的要算作UNIX了。除此之外,还有CP/M 68k、Multiplan这样的OS及其软件。

在这样大的软件里之所以使用C语言,主要有两个原因,即C语言移植性好和硬件控制能力高。如果应用程序用C来描述,则由于C移植性好,应用程序的移植性也就好。此外,由于C的类型检验比较弱,能直接访问物理地址等特点,使得它在直接操作硬件功能时,能实现同汇编语言相同的描述。这就是过去用汇编语言所编写的操作系统可以使用C语言的原因。

由此看来,实际上C所能迅速应用到的领域正是以前汇编语言所应用的领域,因此,C被称作是“便携汇编语言”。它是介于从Ada、Smalltalk到FORTRAN这样的高级语言和汇编语言之间的一种语言。因此,我们可以说,C语言既具有高级语言的特点,又具有汇编语言的特点;它既是成功的系统描述语言,又

是一个实用的程序设计语言。这就是所谓的C语言的双重性。

(3) 程序设计自由度大 C语言程序设计在语言自身的规格上能够相当自由地进行, 故说C语言是水平较低的语言, 也可以说是较为接近汇编语言的一种语言。

由于C语言是水平较低的语言, 故对变量类型、变量范围和存储空间的存在制约就少, 这样, 程序设计就能很自由地进行。

对于程序员来说, 若使用高级语言, 只要按照语法规则来编写, 就能够比较简单地编写出正确的程序, 并能自动地进行错误检验。而C语言, 由于其自由度较大, 错误检验在各个阶段难以进行, 要用它编写出正确的程序是较难的。

这意味着, 同其它语言相比, C语言具有面向专业的特征。

(4) 表示方法简洁

①语句简洁: C语言和PASCAL语言都是由ALGOL语言发展过来的, 是一对孪生姐妹。但C语言要比PASCAL语言表达简洁, 如图1.2所示。

C语言	PASCAL语言
{	begin end
*if (e) S;	if e then S
int i;	var i: integer
int f();	function f(): integer
int a[];	var a: array [...] of integer
int *p;	p: ↑ integer

图1.2 C和PASCAL的比较

②表达式能用赋值语句: 如,

```
while ( (c = getchar( )) != EOF ) putchar(c);
```

这里, 把getchar()的返回值赋给c, 这个值就是c = getchar()的值。

③变量更新表示简单: 在C语言中

```
i = i + 1;
```

可以表示成:

$i += 1;$

这种表示方法也适用于其它运算符。

④变量的自增/自减表示简单:

$i++$ $i--$ (1)

$++i$ $--i$ (2)

注意: (1)和(2)的相同点是运算后*i*都是自动+1/-1;
(1)和(2)的区别是: $++i/--i$ 是先进行加/减1再使用, 而 $i++/i--$ 则是先使用再进行加/减1。

⑤有条件运算符($?:$): 如,

$(x! = 0) ? 1/x : INFINITY$

这一表示的含义是: 若 $(x! = 0)$ 为真, 则求解 $1/x$, 否则值为INFINITY的值。在其它语言里, 要表达上述功能, 必须使用if语句。

⑥关键字少: 各种C处理系统一致使用的关键字有28个, 即:

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

除此之外, 有些系统还使用下面4个关键字:

fortran asm viod enum

加到一起, 才32个。这些关键字都是用小写字母定义的, 使用时, 不能用大写字母书写; 关键字不能挪作他用, 不允许用它们定义变量名、函数名等; 关键字entry还未由任何编译程序实现, 留作将来使用。

⑦宏控制行能指令预处理程序处理宏展开。这里，首先介绍两条，以备使用。

宏定义：

```
#define INFINITY =1
```

这一控制行，使程序中的字符串“INFINITY”用“-1”置换以后，进行编译。这样，在用C语言所编写的程序里，一般不使用整型常量，而使用宏定义。

因为用#define也能展开参数，所以使用宏处理也能描述一些简单的函数，如：

```
#define max(a,b) ((a>b)? a:b)
```

包含文件：

```
#include "stdio.h"
```

这一控制行说明“stdio.h”这一文件被包含。在用C语言所编写的程序里，一个模块内所使用的函数完全可以用另一个文件来描述，分别进行编译，然后再连结起来。也可以使用宏预处理#include把另一个文件包含进来，一起编译。

可以通过#define把共同使用的宏预处理用一个文件来描述，然后在各个文件里用#include读取。这样，在常数需要变更时，只需修改一处即可。

§ 1.2 C 程 序

1. C程序的格式

用C语言所编写的程序，称作C语言源程序，简称C程序（或源程序）。C程序一般由一个或多个函数组成，这些函数可集中放在一个或分散放在几个文件中。为表示这些文件都是C程序文件，各文件都必须以.c结尾，如sum1.c就是一个C程序文件，该文件只含有一个函数main()，如练习1.1所示。

【练习1.1】

```

A> type sum1.c
1  main( )/* ...add of a and b... */
2  {
3  int a,b,sum;
4  a = 123;b = 456;
5  sum=a+b;
6  printf("sum is %d\n",sum);
7  }
A>

```

就该例，对C程序，说明如下：

(1) C程序无行号，这里的行号是为了便于讲解而加的。

(2) sum1.c是文件名。在这里，我们看到C程序文件的结尾是.C。.C即表示C源文件。

(3) /*与*/之间的内容是注释。本例中，它说明main()函数是求解a和b两个自变量的值之和。注释不被编译，只起备忘和帮助阅读的作用。

(4) 在C程序中，其后紧跟()的标识符是函数名。我们已经看到，该程序只有一个函数，即函数main()。这里，main是表示程序的主体部分，即是主程序的约定词。main()可叫主函数。

在执行C程序时，一调用函数名为main()的函数，程序便开始执行；而从main()返回时，程序执行便终止。

在一个完整的C程序中一定要有一个函数main()，不管这个函数放在什么地方，C程序总是从它开始执行。函数main()可以调用其它函数；其它函数之间也可以有调用和被调用的关系；此外，一个函数还可以调用其本身，这叫做递归调用。

(5) 花括号{}相当于ALGOL/PASCAL语言中的关键字begin和end，也相当于PL/1语言里的do和end。它是函数体的界限符。

(6) 3行为说明语句。说明变量a、b和sum的值为整型数(int)。int是integer(整数)的缩写。该说明语句的具体任务是通知计算机以下事项:

①该程序使用了三个变量: a、b和sum;

②这些变量的取值是整数。

这个程序里出现的变量名a、b和sum的含义, 正如 图1.3所示, 是给计算机内存单元起的名字, 各单元内存放着各自的数据, 分别为: 123、456、579。请注意, 所用到的这三个变量名所表示的三个内存单元未必是相邻的。之所以称作变量名, 是因为在这种状态下重新赋值后, 例如:

a=234; b=567; sum=x+y;

则a、b、sum的内容便成为如图1.3(b)所示的情况, 发生了变化, 故把它们叫做变量。

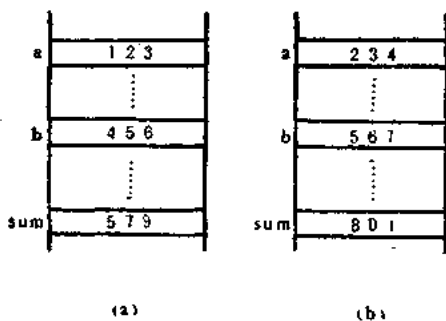


图 1.3 变量的含义

(7) 4、5行中出现的=, 是赋值运算符, 表示把赋值运算符右边的数值或是运算结果赋值给其左边的变量。

(8) 6行中的printf()是按格式输出函数。它不是C语言固有的成分, 而是由宿主系统(如UNIX、CP/M等操作系统)提供的。这里先简单介绍一下这个函数, 以备使用。

%d是printf()函数的最简单的打印格式, 习惯上, 叫转换控制。它指出把后面的相应参数值(这里是sum的值), 以所

需要的最少位数，按10进制打印，前后不留空格。

(9) 语句后面要加；，表示语句结束。带有；号的语句叫简单语句，它是执行单位。在C程序中printf()语句和赋值语句是最基本的语句。若干个简单语句用{}围起来，就构成一个复合语句。复合语句是简单语句的集合，在语法上就象一个简单语句那样使用。凡是能用简单语句的地方，都能使用复合语句。

在C程序里，不是每个简单语句必须占据一行。一行中可以有若干个语句，如练习1.1中第4行所示；但每个简单语句后，必须要有一个分号(；)。

有一点要特别注意，即花括号}前一个语句末的分号不能少；而在}后面又不能有分号。在这一点上，C语言和PASCAL语言不同。在PASCAL中end前一个语句不能有分号，有了则看作end和其前一个语句之间还有一个空语句。

(10) 该程序的执行结果是：

```
A>sum1
sum is 579
```

这里必须强调的是，在C程序里提倡把一个程序分割为若干个函数来编写，又提倡把各个函数分别放在不同的文件里。这主要是考虑到一个有实用价值的程序往往需要反复进行修改、调试。如果把程序编制成一个大程序，尽管有时看起来还算紧凑、简洁，但只要修改一处，也要重新编译一次整个程序，这在时间上会造成很大的浪费。如果我们把一个程序分成若干个函数来编写，并把这些函数分别放在不同的文件里，那么我们修改某个函数，就只需重新编译一下这个函数所在的文件，而其它未被修改的函数所在的文件就无须重新编译，这样，编译时间会大大减少。待各个文件的编译都通过之后，通过函数调用，就可以把这些函数链接成一个完整的程序来执行。

还需要指出的是，程序中函数彼此间的排列顺序如何是没有关系的，当然是尽可能整理成便于理解整个程序的结构顺序为