



郭 荷 清 编 著

C程序设计语言

广东高等教育出版社出版

C 程 序 设 计 语 言

郭荷清编著

广东高等教育出版社

C 程序设计语言

郭 荷 清

广东高等教育出版社出版

广东省农垦总局印刷厂印刷

787×1092毫米32开本8.3125印张 180,000字

1984年11月第1版 1984年11月第一次印刷

印数1——30,000册

书号 15343·4 定价1.70元

前 言

C语言是七十年代初期由美国贝尔实验室的Dennis Ritchie博士等设计的，它是在UNIX系统上研制成功的，并且在1973年将原来以汇编语言写的UNIX系统改用C语言重写，除与硬件有密切联系的10%的内容仍用汇编语言描述之外，其余都是用C语言书写的。另外，C编译程序则完全用C语言编写，它具有高度的可移植性，因为对于不同的计算机上的C编译程序有80%的代码是公共的。要为一台计算机写一个编译程序相对来说是比较容易的，这为C语言的移植能力提供了方便，尤其是在当前，C语言已经出现在微型机系统中，虽然C语言是在UNIX系统上研制的，但是它并没有被束缚在特定的机器或操作系统上，目前在许多流行的微型计算机上例如：PDP—11/23，MC68000，IBMPCXT，APPLE IIe，CROMEMCO等机器上都配有C编译程序，可以使用C语言。而它们所运行的操作系统除UNIX系统外，还有UNIX系统的变种操作系统，CP/M等操作系统。C语言已越来越受到人们的重视。

C语言是一种通用的程序设计语言，它有着经济实用的表达式，先进的控制流和数据结构，以及丰富的运算符。它最初的设计目的是为了开发UNIX系统，试图以一种较高级的语言来取代编写系统软件惯用的汇编语言，它在一般高级语言的基础上又吸取了汇编语言在实践中有益的经验。所以它既有高级语言的特点，可以不依赖机器进行编程，而且C程序的可读性好。同时它又具有汇编语言的特点，如：位操作

等功能，并且较多地反映了底层硬件的处理功能，容易模块化，更重要的是编译出来的目标代码功效高（仅次于汇编语言的目标代码），提高了运行速度，对程序的调试工作可以大大减少，缩短了软件的研制周期，C语言不仅在系统程序设计方面发挥了巨大的威力，而且在应用程序设计方面也取得了很好的效果。鉴于C语言的特点，尤其在促进微型机系统的通用化，以及能够方便的移植开发软件，因此，它在软件开发技术中可以发挥很大的作用，本书是为这种目的而编写的。

全书共分六章：第一章研究基本的数据类型、运算符和表达式；第二章研究C语言的控制流语句；第三章讨论数组、指针和地址的运算；第四章讨论函数和程序结构；第五章讨论结构和联合的内容；第六章叙述“标准I/O库”，它是一组函数，为C程序提供一系列标准I/O系统。最后部分是C语言参考手册。

这六章是C语言的基本部分，从读者学习的角度出发，由浅入深的安排内容，并有大量的例子说明各部分的功能，经过这六章的学习之后，就可独立编制程序。

本书的附录一，对于想了解C语言的编辑、编译和运行或初次上机的读者将有帮助；附录二，专门讨论与UNIX接口问题，对学习C语言作为熟悉UNIX系统的工具的读者是很有用的；附录三，汇总了C语言的基本语法，并对一些语法作了较严密的定义。

本书在定稿过程中，暨南大学吴恭顺副教授提出了宝贵的意见，在此谨致谢意。

编著者

1984年9月

目 录

前 言

第一章 类型、运算符和表达式	1
§ 1—1 变量.....	1
§ 1—2 常数.....	3
§ 1—3 运算符.....	8
§ 1—4 运算符的优先级和结合性规则.....	16
§ 1—5 类型转换.....	17
§ 1—6 表达式.....	21
习 题	23
第二章 控制流语句	24
§ 2—1 语句和复合语句.....	24
§ 2—2 赋值语句.....	25
§ 2—3 条件语句.....	25
§ 2—4 循环语句.....	28
§ 2—5 开关语句.....	39
§ 2—6 中止语句.....	45
§ 2—7 继续语句.....	49
§ 2—8 转移语句.....	51
§ 2—9 空语句.....	52
§ 2—10 返回语句.....	53
习 题	54

第三章 数组和指针.....56

§ 3—1	数组.....	56
§ 3—2	指针.....	66
§ 3—3	指针和数组.....	69
§ 3—4	指针数组.....	73
§ 3—5	指针和数组的区别.....	79
习 题		83

第四章 函数.....84

§ 4—1	函数的形式.....	84
§ 4—2	函数调用——参数传递和调用返回.....	88
§ 4—3	存贮类.....	94
§ 4—4	程序结构	109
§ 4—5	初始化	111
§ 4—6	函数的递归调用	114
§ 4—7	C预处理 程序	117
§ 4—8	指向函数的指针	120
习 题		124

第五章 结构类型126

§ 5—1	基本概念	126
§ 5—2	指向结构的指针	131
§ 5—3	结构数组	134
§ 5—4	引用自身的结构	143
§ 5—5	字位段	147
§ 5—6	联合	150

习 题	152
第六章 库存函数	154
§ 6—1 输入函数和输出函数	154
§ 6—2 字符串处理函数	170
§ 6—3 字符转换成整数函数	174
§ 6—4 文件存取	176
§ 6—5 错误处理及出口	180
§ 6—6 行输入和输出	181
§ 6—7 其它函数	183
附录一 C语言程序的编辑、编译和运行	185
附录二 与UNIX 接口	192
附录三 C语言参考手册	213

第一章 类型、运算符和表达式

§ 1—1 变量

在C语言程序中，所有的变量必须加以说明。虽然有时这种说明可以根据上下文的内容隐含地进行。说明必须在可执行的语句之前。

最简单的说明的语法规则是：

类型 变量表；

例如：int a, b, c, sum;

它说明了变量a, b, c, sum是整数型。其中int是类型，后面是变量表，各变量之间用逗号“,”分开，最后用分号“;”结束。

在C语言中有四种基本变量类型：

int 整型

char 一个字节的字符

float 单精度浮点

double 双精度浮点

还有四种由这些基本类型构成的变量类型：

array 数组

structure 结构

pointer 指针

function 函数

此外，对于整型数还可以加限定词：short（短），long（长），unsigned（无符号）。short和long表示不同长度的整数。unsigned数总是遵守模 2^n 的运算规则，无符号数总是正的。带有限定词的变量说明如：

short int x; 表示短整数 x
 long int y; 表示长整数 y;
 unsigned int z; 表示无符号整数 z
 其中int可以省略。

这些变量的精度依赖于所用的机器。下表给出几种机器上的精度值。

机型	DEC PDP-11	Honeywell6000	IBM 370	Interdata8/32
类型	ASCII	ASCII	EBCDIC	ASCII
char	8 位	9 位	8 位	8 位
int	16位	36位	32位	32位
short	16位	36位	16位	16位
long	32位	36位	32位	32位
float	32位	36位	32位	32位
double	64位	72位	64位	64位

变量名由1个到8个字符构成，它们由字母A~Z, a~z 和数字0~9及底线字符“_”组成。但开始的第一个字符必须是字母。按C语言的传统习惯，变量名用小写字母，而符号常数全部用大写字母。函数名和外部变量名则由6个字符组成。此外，下面这些字将保留作为关键字，不能用作

变量名（他们必须小写）：

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

§ 1—2 常数

一、数字常数

1. 八进制数以前导零开头

例如：037

2. 十六进制数以前导“0x”开头

例如：0x1f

3. 十进制数以数字开头

例如：31

以上常数后面可以带字母L，使之成为长整常数。如：31L，0x1fL，037L。

二、浮点常数

例如：0.37e²

三、字符常数

1. 字符常数是使用单引号括起来的单一字符，一个字

符常数的值是该字符在机器的字符集中的数字值。

例如：字符零（即“0”）在ASCII字符集中的值是48。

2. 不可显示的字符可以用特殊的字符。

例如：“\n” 换行

“\t” 制表

“\0” 空白

“\” 反斜线本身

“\b” 退一格

四、字符串常数

字符串常数是用双引号括起来的零个或多个字符序列。

例如：“ ” 零字符串

“ABC” 字符串

这里的双引号不算是字符串的一部分，而只充当界限符。字符串是一个数组，其元素为单个字符。编译程序自动地在每个字符串末尾放上字符\0，使程序能方便地找到它的结尾。\0要占一个内存的字符单元。所以，一个字符串所需的物理存贮空间比双引号之间的字符个数多一个字符单元。

例1—1. 已知三个整数1, 2, 3求和并打印出结果，下面用C语言写出程序为：

```
main ( ) {  
    int a,b,c,sum;  
    a = 1; b = 2; c = 3;  
    sum = a + b + c;  
    printf ("sum is%d\n", sum);  
}
```

上例第一行中的main是一个特殊函数，所有C程序都必须有一个main函数，程序的执行从main的第一个语句开始。main常常要引用其它函数来执行它的作业，有些函数来自同一程序，而另一些函数则来自以前写好的函数库。函数间数据通讯的一种方法是通过参数来实现。参数跟在函数名后面，并用圆括号括起来。这里main是无参函数，括号内是空的，就用main ()表示。花括号 { }把构成函数的语句括起来，它类似于Algol或Pascal中begin—end的作用。

第二行int a, b, c, sum; 是变量说明语句，其中a, b, c, sum都是变量，用int来说明它们的类型，都是整数变量。

第三行a = 1; b = 2; c = 3; 分别是三个赋值语句。C程序的格式是很自由的。我们可以把几个语句放在一行，也可以把一个语句分开放在几行中。要注意，这种分放可以发生在任何操作符或变量之间，不能在一个名字或一个操作符的中间分开。但是，构成一个语句，作为语句结束符的分号“;”，绝对不能忘记。

第四行sum = a + b + c; 是算术运算语句。

第五行printf ("sum is %d\n", sum); 是输出语句。它要求按一定格式把和数sum按十进制数输出，本例的打印结果为：

```
sum is 6
```

括在双引号“.....”里的字符序列称为字符串，在打印时照样打出，而%d字符是将变量（本例是sum）的数值用十进制打印，在字符串中最后的“\n”是C语言的换行符，打印完后移到下一行的最左边。要注意\n只代表一个字符。

printf实际上是一个通用格式转换函数，为C程序正常存取的标准子程序库的一部分，它能识别变量的输出方式，其符号和含义为：

%d 十进制数
%o 八进制数
%x 十六进制数
%c 字符
%s 字符串
%% %本身
%f 浮点数

例 1—2. 输出语句练习

```
n = 511
printf ("what is the value of %d in octal?
", n) ;
printf ("\%s! %d decimal is %o octal\n",
        "right", n, n) ;
```

结果是：

```
what is the value of 511 in octal? right! 511
decimal is 777 octal
```

注意：在第一个输出行的末尾没有放入换行标志“\n”，所以在继续调用printf时，紧接第一个输出行后面继续输出结果，直到遇到第二个输出行的“\n”时才结束。

在输出语句中，输出方式表示符和输出量要一一对应，输出量跟在双引号后面，并用逗号“，”分隔开，如第二个输出行中，%s字符串输出表示符与“right”输出量相对应，%d与n相对应，%o与n相对应。

此外，在%d，%f输出方式表示符中可以加入参数，例

如%4d表示十进数要占至少四个字符的宽度；%4.0f表示浮点数要占4个字符的宽度，小数点后不带数字；%.2f表示小数点后要有两个位置，但对其宽度不加限制。

另外，输出量可以用该类型的表达式表示。

例1—3. 文件复制

```
main ( )
{
    int c;
    c=getchar ( ) ;
    while (c!=EOF) {
        putchar (c) ;
        c=getchar ( ) ;
    }
}
```

和printf一样，getchar和putchar都不是C语言的一部分，它们是C语言标准的I/O库函数，可以普遍使用。有了getchar和putchar后，无需对I/O有更多的了解，就可以写出大量有用的代码。

例1—3中的前三行在例1—1中已有说明，不再重复。

第四行c=getchar ()中，getchar ()是输入函数，每次调用它时，就从标准输入设备（通常用户终端）读入一个字符，并且送回该字符作为输入函数的值。此时，变量c中包含了输入的一个字符。

```
    第五行while (c!=EOF) {
        ...
    }
```

while语句是C语言中的循环语句。检查圆括号中的条

件是否为真，如果为真，则执行循环体（由花括号 { } 括起来的所有语句），否则循环结束，并继续执行循环体之后的语句。在本程序中，它后面没有语句了，所以程序结束。

关系运算符 “!=” 表示“不等于”。

文件结束标志 EOF 表示已读到文件终点的字符，即 end-of-file 的缩写。

第六行 putchar (c) 是输出函数，在本程序中是输出刚读入的字符。

§ 1—3 运算符

C语言使用的运算符比较多，由于它采用特有的运算符，使程序文本的书写更为简练，下面介绍各种类型的运算符。

一、算术运算符

+ 加

- 减

* 乘

/ 除

% 除后取余数

例： `sum = a + b - c`

例： `x % y`

产生 `x` 被 `y` 除所得的余数。如果能除尽，则其值为 0。

二、关系运算符

< 小于

<= 小于或等于

> 大于

>= 大于或等于

`==` 等于

`!=` 不等于

关系运算符是自左向右组合，若规定的关系式不成立，会产生零；若规定的关系式成立，则会产生1。结果类型是整型，并执行通常的算术转换。

三、逻辑运算符

`&&` 逻辑与

`||` 逻辑或

`&&`运算符自左向右组合。若两个操作数均非零就返回1，否则返回零。它与`&`不同，`&&`保证自左向右计算，而且若第一个操作数为零，就不计算第二个操作数。两个操作数不必是同类型的，但每一个必须是基本型或者是指针，结果总是`int`型。

`||`运算符自左向右组合，若有一个操作数非零则结果为1，否则为零。与`|`不同，`||`保证自左向右计算。而且，若第一个操作数非零则不去计算第二个操作数。各个操作数不必是同类型的，但每一个都必须是基本型或是指针，结果总是`int`型。

例：某年数如果能被4除尽，但不能被100除尽，这年就是闰年，若能被400除尽也是闰年。

其条件表达式可写成：

```
if (year%4 == 0 && year%100 != 0 || year%400  
    == 0)
```

这年是闰年

否则

这年不是闰年

注意：`&&`的优先级高于`||`，但都比关系运算符、算术