C语言精要

唐龙 徐玉华 编著

清华大学出版社

C语言精要 唐龙 徐玉华 编著

C语言精要

唐　龙　徐玉华　编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

C 语言是十分重要的计算机程序设计语言。为使各种专业各种层次的读者能尽快掌握 C 语言，本书简明扼要地按一般教材的编写方式讲解了 C 语言，内容全面，实例丰富，详略得当。

本书适于高校非计算机专业学生学习 C 语言使用，亦可用作各类计算机培训班教材。

图书在版编目(CIP)数据

C 语言精要/唐龙等编著．—北京：清华大学出版社，1995.2
ISBN 7-302-01680-1

Ⅰ.C… Ⅱ.唐… Ⅲ.C 语言 Ⅳ.TP312C

中国版本图书馆 CIP 数据核字(94)第 13729 号

出版者：清华大学出版社（北京清华大学校内，邮编 100084）
印刷者：北京市海淀区清华园印刷厂
发行者：新华书店总店科技发行所
开　本：787×1092 1/16　印张：10.75　字数：251 千字
版　次：1995 年 2 月 第 1 版　1995 年 2 月 第 1 次印刷
书　号：ISBN 7-302-01680-1/TP・726
印　数：0001—8000
定　价：8.50 元

编者的话

C语言的学习、应用，从来都是计算机教学与实践的关键和难点。特别在计算机高速发展和普及的今天，越来越多各行各业的人们正自觉不自觉地带着各自不同的专业背景、面向着不同的专业方向，一步步逼近到计算机的内核，对机器、对自己都提出了越来越高的要求。此时，彻底、迅速地掌握一门计算机语言，已成为当务之急。而对无论置身于哪一专业领域的计算机爱好者来说，无论您将面向文字的高级处理还是数据管理，无论您将主攻自动控制还是CAD的高级应用，C语言都当成为您首选的计算机语言。甚至可以这样说，如果您永远不懂C语言的话，那您将永远被拒绝在计算机专业技术的门外。

本书正是为各行各业，当然也包括计算机本专业的读者编写的一本简明C语言读本。书中以通俗易懂，清晰精炼的叙述，完整地讲解了C语言的语法、程序结构、使用方法及一些关键的应用技巧。

本书作者多年从事C语言的培训、教学工作，为学校及社会上各种专业背景的学生以本书的原始讲义为教材讲授了多遍C语言，深谙学员心理及接受能力。本书适于作为高校非计算机专业的C语言教材，对各类计算机培训班来说，本书更是一本详略得当、恰到好处的教材。

目　录

第1章 概 述

§1.1 C 语言的由来和特点

1. C 语言是一种极有生命力的计算机程序设计语言

C 语言是一种有强大生命力的计算机程序设计语言。它比 BASIC,ALGOL,FORTRAN,PASCAL,PL/1,COBOL 等语言出现得都要晚。1972 年 C 语言才正式问世。此后,1978,1983 和 1987 年几次修订过标准化版本。如今,它已成为一种较成熟的编程语言,也是现代计算机科学中最广泛使用的、必不可少的一种语言。

当然,C 语言的产生也不是偶然的,就其本身的孕育发展而言,也有一个过程。

1967 年,首先由 Martin Richards 开发出 BCPL 语言。作为软件人员用于开发系统软件的描述语言,BCPL 语言的突出特点是:

(1) 结构化的程序设计;

(2) 直接处理与机器本身数据类型相近的数据;

(3) 具有与内存地址对应的指针处理方式。

1970 年,Ken Thompson 继承并发展了 BCPL 语言的上述特点,设计实现了 B 语言。当时,美国 DEC 公司的 PDP-7 小型机 UNIX 操作系统,就是使用 B 语言开发的。

1972 年,美国 Bell Lab. 在 PDP-11 小型机的 UNIX 操作系统开发中,Dennis M. Ritchie 和 Brian W. Kernighan 对 B 语言做了进一步的充实和完善,正式推出了 C 语言。

2. C 语言的特点

如果将 C 与 FORTRAN、PASCAL 等作比较,不难发现它吸收了其他计算机语言的某些优点,抛弃了某些缺点。当然,每一种编程语言都有其特点,其中某些优缺点只能是同时并存,或者说只是立足于不同观点的观察而言。

C 语言有如下特点:

(1) C 语言是介于汇编语言与高级语言之间的一种描述程序语言,也有人称之为中级语言。这是很自然的,因为它正是由写操作系统(UNIX)的需要而发展起来的,所以,它比较接近于汇编语言,有着面向硬件系统、便于直接访问硬件的功能;同时,为便于加快开发速度,提高工作效率,它又具有类似于高级语言面向用户,容易记忆、便于书写和阅读的优点。

(2) C 语言是一种灵巧的、结构化的、模块化的强有力的软件设计语言。它只有很少的核心,只使用很少的关键字。通过明确的控制结构,丰富的数据类型和运算操作、函数库以及模块化的编程手段,使 C 语言成为一种强有力的编程语言。

(3) C 语言是一种可移植性很强的语言。这体现在两方面：其一，I/O 功能通过调用 I/O 函数库来实现，而这些函数是系统提供的独立于 C 语言的程序模块库，这样 C 语言本身可不依附于机器硬件系统。其二，一种机型的 C 语言核心编译器可以很小，系统的实用性部分和预处理部分均与机器无关，使它从一个系统到另一系统的移植改写很容易实现。

(4) C 语言在编程上提供了较大的自由度，与之相伴的是非约束性，这也造成了某些困扰。C 语言的格式比 FORTRAN 要自由得多。C 语言的预处理语句，例如，#define，#include，#if 和 #else 等所提供的宏定义，对外部文件的使用以及条件编译等措施均有利于提高开发效率，但也给不熟练的开发者增加了出错的可能性。

3. C 语言程序的生成和使用过程

在 UNIX 操作系统支持下 C 语言程序的生成和使用主要过程，如图 1.1 所示。

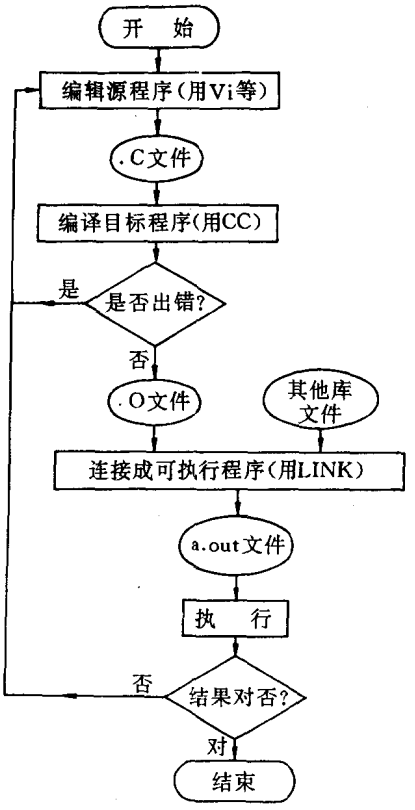

图 1.1

§1.2 C语言程序的基本形式、标识符和保留字

1. C语言程序的基本形式

C语言程序都由一个或多个函数(Function)构成。一个C程序至少必须存在一个函数“main()”。它是程序运行开始时调用的一个函数。它表明该程序完成动作的轮廓。C语言程序的基本形式如下：

```
main()
{
    变量说明语句;
    执行语句;
}
```

main()为主函数。执行语句中，可有其他函数，但不能用main为函数名。许多常用的函数做成标准函数与C编译器一起提供给用户，这就是标准库函数。

例1.1：

```
/* This program is sum of two integer and displays the results */
    main()
    {
       int  a,b,sum;
       a=123;
       b=456;
       sum=a+b;
       printf("The sum of %d and %d is %d\n", a,b,sum);
    }
```

此实例中，只有一个函数，即主函数main()。a,b和sum都是变量。第一个语句就是说明这些变量的语句，说明它们都是整数(即int)型的变量。后四个语句就是执行语句。所有这些语句都放在左右大括号之内，各语句之间以分号“;”断开。

2. 标识符

C语言中所使用的每个函数和变量都应有唯一的名称，这样才能被识别和使用。通常，这种函数和变量名称用一串字符表达，称为标识符。在C语言中，使用的标识符有严格限制，即：

(1) 必须以字母或下划线开头；
(2) 必须由字母，数字或下划线组成；
(3) 大小写字母是有区别的；
(4) 不允许用一些保留字(或叫关键字)。

例 1.2：正确的函数或变量名称的实例：

_abc

Tsx7

Lev_5

例 1.3：错误的函数或变量名称的实例：

3H

sUM $

3. 保留字(即关键字)

在C语言中保留字或关键字并不太多，原先规定有 28 个，新标准规定改为 32 个，如下所示。

auto	default	extern	long	static	void②
break	do	for	register	struct	volatile②
case	double	float	return	switch	while
char	enum②	goto	sizeof	typedef	
continue	else	if	signed②	union	
const②	entry①	int	short	unsigned	

[注] ① 表示旧标准所用的，新标准中已不用了。

② 表示新标准所增加的。

此外，在某些版本中，还可能包含其他几个保留字或关键字，例如：ada，asm，fortran 和 pascal 等。

另有若干非关键字，也不可随意使用，例如：define，undef，include，ifdef，ifndef，endif 和 line 等均为预处理器使用的一些字。

§1.3 基本数据类型

1. C 语言可以使用多种数据类型

(1) 基本类型

- 整数类型
- 实数类型(浮点类型)：
  - 单精度浮点型
  - 双精度浮点型
- 字符类型
- 枚举类型

(2) 构造类型

- 数组类型
- 结构类型

· 联合类型

(3) 指针类型

(4) 空类型

本章先介绍三种基本类型，其他在后面章节中陆续介绍。

数据有常量和变量之分。常量是指在运算过程中，其值不变的量，可以是一个具体的值，也可以定义一个标识符来代表，即符号常量。变量的值在运算过程可以改变。如前节所述，每个变量都应有一个唯一的名称，并根据其类型在存储器中占一定的存储单元，以便存放其值。

2. 整数型(即 integer——int 型)

整数型常量不仅可用十进制表示，也允许用八进制或十六进制表示，例如：

32　　十进制数

032 即 26　　八进制数　　以 0 为先导，以 0～7 构成。

0747　即 (7×8+4)×8+7=487

0x32 即 50　　十六进制数　　以 0x 为先导，以 0～9 和 A～F 构成。

0x2C5 即 (2×16+12)×16+5=709

整数类型又可细分成不同长短的类型，应加上类型修饰符来构成，即：

short int　　可简化为 short

long int　　可简化为 long

unsigned int　　可简化为 unsigned

(1) 短整数 short，其长度通常是 2 byte (即 16bit)，取值范围为：

-2^{15}～ $2^{15}-1$ 即：−32768～+32767

通常，int 就是 short，存放在一个字中，即占 2 byte。

(2) 长整数 long，其长度通常是 4 byte (即 32bit)，取值范围为：

-2^{31}～$2^{31}-1$ 即：−2147483648～+2147483647

(3) 无符号整数 unsigned，其长度通常是 2 byte (即 16bit)，取值范围为：

0～$2^{16}-1$ 即：　0～+65535

通常，unsigned short 也就是 unsigned int。

(4) 无符号长整数 unsigned long，其长度通常是 4 byte (即 32bit)，取值范围：

0～$2^{32}-1$ 即：　0～4294967295 (≈4 billion)

请注意：不同机器上数据类型长短可能不一样，例如，在 IBM PC 机或 PDP-11 机上，unsigned int 等同于 unsigned short，然而，在 IBM 370 机或 VAX-11 机上，unsigned int 则等同于 unsigned long。

3. 浮点类型(即 float 型)

实数类型也叫浮点类型。通常，它可包含有整数部分和小数部分，例如：

PI 等同于 3..1415926536

0.012 等同于.012

5.0 等同于 5.

和日常习惯一样，小数点(.)的左边为整数部分，右边为小数部分。

科学计数法中，则用“尾数+e+指数”来表示浮点数值，e 即 exponent(指数)例如：

3.4e5=340000.0

其中，3.4 就是尾数，含有整数部分(integer part)为 3 和小数部分(fractional part)为 4，而 5 就是指数部分(exponentical part)。尾数和指数也都有可能为负值，例如：

−1.23e4

12.34567e−8

−89e−12

一般计算机中，标准的方法是用 1 byte 的指数加 3 byte 的尾数来表示浮点数，而且，尾数经过规格化，使之小于 1 并大于或等于 0.5(等于 0 除外)，也就是使尾数部分的二进制最高位总是 1。

1 byte 的指数，除符号位外，有效值是 7 bit，$2^7-1=127$，即为指数的最大值。在机器中，总是二进制表示，$2^{127}\approx1.7\times10^{38}$。所以，浮点类型的取值范围是：

$10^{-38}\sim 10^{38}$

有效数位数取决于 3byte 的尾数，即除符号位外，有 23 bit，$2^{23}\approx8.4\times10^6$，即相当于 7 位十进制的有效数表示尾数。

双精度 double 用 8 byte 表示一个浮点数，指数仍是 1 byte，尾数 7 byte，因此，double 的取值范围也是：

$10^{-38}\sim 10^{38}$

而尾数则有 $7\times8-1=55$bit，$2^{55}=3.6\times10^{16}$，即相当于 16 位十进制有效数表示尾数。

4. 字符类型(即 character——char 型)

字符类型的数据代表一个字符，由一对单引号将字符括起来，表示的是该字符在 ASCII 码表中的代码值，例如：

'a'　　即 97

'A'　　即 65

它们占 1 byte。

ASCII 码表中的某些控制字符不可显示，则通过加反斜线“\”的转义字符表示，例如：

'\0'表示 null(空)	即 0
'\t'表示 tab(制表)	即 9
'\n'表示 new line(新行或换行)	即 10
'\r'表示 return(回车)	即 13
'\\'表示\(反斜线)	即 92

注意：单引号括起来的字符是指一个 char 类型的数值，而不是字符串，后面我们会讲到双引号括起来的字符序列，才是字符串。

§1.4 算术表达式

1. C 语言中算术运算符

算术表达式由变量、常量及算术运算符构成。在 C 语言中算术运算符有：

＋ － * / % －－ ＋＋

＋，－，* 和/为四则运算符，它们和日常概念没什么区别，其中 * 和/优先于＋和－。

% 为取模(Modulus) 运算符，是针对整数运算，即取整数除法之后，所得到的余数，例如：

10%3=1　　即 10 对 3 取模，结果为 1。

13%8=5　　即 13 对 8 取模，结果为 5。

－－ 为自减 1，＋＋ 为自增 1。

特别注意，n＋＋或＋＋n 都是变量 n 自增 1，最终结果均与 n=n+1 一样。但处理过程却有所区别。＋＋n，表示 n 先自增 1，然后进到具体的式子中运算；n＋＋，则 n 本身先进入式中运算，最后 n 再增 1。

例如：已知 n=2，则

m=＋＋n；结果为：m=3，n=3

m=n＋＋；结果为：m=2，n=3

n－－与－－n 与此类似。

2. 数据类型与运算结果的关系。

(1) 同类型数据运算结果仍保持原数据类型

整型数的除法得到的结果仍是整型数，小数部分将被去掉，例如：

5/2=2

而不是 2.5。浮点数的除法得到的仍是浮点数，例如：

5.0/2.0=2.5

(2) 不同数据类型混合运算，精度低的类型往精度高的类型转换后，再做运算。这样，可保证运算结果不损失精度。例如：

5.0/2=2.5

§1.5 C语言的基本语句

1. C语句的特点

(1)所有C语句都以“;”分号结尾。一条语句可以不止一行，不必加续行符，只根据“;”来确定语句结束。两条或多条语句也可写在同一行，只要有“;”分开就可以。

(2) 语句可从任一列位置开始，每行开头，有多少空格都可以，但为了可读性好，通常习惯还是按一定规律缩进。

2. 变量说明语句

变量说明语句的主要作用就是定义变量类型，其格式是：

类型说明符 变量1[,变量2,...];

例如：

```
int number;
char a,b,c;
float t;$ $ $ $
float t;
```

3. 赋值语句

赋值语句是将常量或算术表达式的运算结果赋给变量，其格式是：

变量名=常量或算术表达式;

例如：

```
int number;
number=10;
```

例1.4：

```
main( )
{
  int n1,n2,n3;
  int total;
  n1=1;
  n2=2;
  n3=3;
  total=n1+n2+n3;
}
```

4. 基本输入输出语句

printf( )和 scanf( )是C语言的基本输入输出函数，都放在标准函数库中，为了使用它们，应在程序开头加上：

```
#include    <stdio.h>
```

基本输入输出语句就是直接调用这两个基本输入输出函数。

(1) 输出语句

一般格式是：

```
printf(控制串[,表达式1, ..., 表达式n]);
```

控制串(或叫格式串)是用双引号括起来的输出格式控制说明。控制串中每一个变量都应当与后面相应的某个表达式对应。

例1.5：

```
printf("Welcome!");
```

结果屏幕上显示：

```
Welcome!
```

例1.6：

```
printf("Welcome\n");
```

与前例不同是后面加了一个换行符。

例1.7：

```
int number=10;
printf("The number is %d\n", number);
```

应显示：

```
 The number is 10
```

例1.8：

```
float value1, value2, value3;
value1=2.3;
value2=4.5;
value3=6.7;
printf("The average of %f and %f and %f is %f\n", value1, value2,
        value3, (value1+value2+value3)/3.0);
```

应显示：

```
The average of 2.3 and 4.5 and 6.7 is 4.5
```

%d 和%f 表示后面要显示的数据类型，且一一对应。%d 表示要显示整型数，%f 表

示要显示浮点型数。

(2) 输入语句

一般格式是：

scanf(控制串，地址表达式 1[,地址表达式 2,...,地址表达式 n])；

控制串(或叫格式串)是用双引号括起来的输入格式控制说明。后面的每个地址表达式的值，应当对应于前面控制串中某一个格式变量的地址。

例 1.9：

```
int number;
scanf("%d", &number);
```

其中，%d 表示应以整型格式输入，&number 表示指向 number 的地址。

例 1.10：

```
float average;
scanf("%f", &average);
```

其中，%f 表示应以浮点型格式输入，&average 表示指向 average 的地址。

4. 注释

程序中的注释部分用/* 和 */括起来，可加在程序中的任何地方，以便加强可读性。也可为调试修改而暂时“删去”程序中某些部分。

例 1.11：

```
/* more arithmetic operations */
  #include   <stdio.h>
  main ( )
  {
     int a=25;
     int b=2;
     printf("a/b*b=%d\n", a/b*b);
     printf("-a=%d\n", -a);
     printf("a+b++=%d\n", a+b++);
  }
```

注释语句中不能再加注释语句。