

C 语言程序设计概述

计算机处理问题是由程序来控制的。程序是人们根据解决问题的思路,利用计算机程序设计语言编制的能完成一定功能的指令序列。程序设计是指用计算机程序设计语言编制计算机程序的过程。

C 语言是一种计算机程序设计语言,起源于 20 世纪 70 年代,最初用于编写 UNIX 操作系统,后来由于 C 语言强大的功能及可移植性,使 C 语言迅速得到推广,并成为世界上应用最为广泛的程序设计语言之一。

本章通过几个简单的例子来认识 C 语言,了解 C 语言的结构特点、书写格式以及在 Dev-C++ 集成环境和 Visual Studio 2019 集成环境下调试 C 语言程序的方法。

1.1 C 程序简介

为了说明 C 程序的结构特点以及书写格式,下面通过几个例子来认识一下 C 程序。

【例 1-1】 在屏幕上输出一行信息。

```
#include <stdio.h>           //编译预处理命令
int main(void)               //主函数
{                             //函数体开始
    printf("This is my first C Program!\n"); //在屏幕上输出信息
    return 0;
}                             //函数体结束
```

运行结果为:

```
This is my first C Program!
```

程序说明:

(1) 程序的第 1 行是 C 语言中以 # 开头的编译预处理命令,以 .h 为扩展名的文件称为头文件,通过文件包含命令 #include 将头文件 stdio.h 包含进自己编写的 C 程序中。所谓头文件是系统内置的已经编写好的程序,用户通过文件包含命令实现对头文件的调用,头文件用尖括号 <> 或双引号 " " 括起来。此处的 stdio.h 是标准的输入/输出函数头文件。若要在程序中实现输入/输出,需使用系统内置的库函数,如该程序中的 printf() 函数,则必须在程序的开始处写上预处理命令 #include <stdio.h> 或 #include "stdio.h",详见附录 B。

(2) 程序的第 2~第 6 行是 C 程序主函数 main() 的定义。main 前面的 int 表明该函

数将返回一个整数值。圆括号中的 void 表明没有给 main 函数传递任何数据。在 C 语言程序中,必须有而且只能有 1 个 main() 主函数,C 程序的执行总是从 main() 函数开始。从{开始,到}结束的部分称为函数体。第 4 行 printf() 函数的功能是在屏幕上输出信息,双引号内的\n 表示换行,即信息在屏幕上输出后,光标定位到下一行。最后的分号“;”是 C 程序语句结束的标志。

(3) 在 C 语言中,以 /* 开头、*/ 结束的内容或以 // 开头的内容是程序的注释。注释是对代码的“提示”,注释可以出现在程序的任何位置,用以帮助理解程序。运行程序时,注释部分将不被执行。一般地,程序块的注释采用 /* ... */ ,行注释采用 “//...”。

(4) 函数体中最后一条语句“return 0;”有两个作用:一是使 main 函数终止(从而结束程序);二是指出 main 函数的返回值是 0,这个值表明程序正常终止。如果 main 函数的末尾没有 return 语句,程序仍然能终止,但是,许多编译器会产生一条警告信息。

【例 1-2】 从键盘输入两个整数,求它们的乘积。

```
#include <stdio.h>
int main(void)
{
    int a,b,cj;                //定义 3 个整型变量
    printf("Please Input Two Integers:\n"); //在屏幕上输出提示信息
    scanf("a=%d,b=%d",&a,&b);      //从键盘输入两个整数分别放入变量 a 和 b 中
    cj=a*b;                    //将 a 和 b 的乘积赋给变量 cj
    printf("cj=%d\n",cj);        //在屏幕上输出乘积 cj 的值
    return 0;
}
```

运行结果为:

```
Please Input Two Integers:
a=7,b=9 ↵
cj=63
```

说明:

(1) 程序的第 4 行是变量的定义语句。C 语言用变量来存放数据,此处用关键字 int 定义 3 个整型变量 a、b、cj,表示这 3 个变量中可以存放整数。在定义变量时,多个变量之间以逗号“,”分隔。

(2) 程序的第 6 行 scanf() 函数的功能是从键盘输入两个整数分别放入变量 a 和 b 中,其中 %d 是格式控制符,表示从键盘输入数据的类型是十进制整数;& 是取地址符,表示从键盘输入的数放到 & 符号后面变量所对应的存储地址中。假如从键盘输入的两个数是 7 和 9,则此处 scanf() 函数的输入方式为: a=7, b=9。注意:双引号中普通字符原样输入。

(3) 程序的第 7 行是赋值语句,“=”是赋值运算符,其功能是把“=”右边表达式 a * b 的乘积值赋给左边的变量 cj。

(4) 程序的第 8 行功能是在屏幕上输出乘积 cj 的值,注意双引号中的 cj= 是普通字符原样输出。%d 表示输出数据的类型是十进制整数,后面 cj 的值对应输出在 %d 的

位置。

【例 1-3】 从键盘输入两个实数,比较它们的大小,在屏幕上输出其中较小的数。

```
#include <stdio.h>
float min1(float x,float y);           //函数声明语句
int main(void)
{
    float a,b,min;                     //定义 3 个 float 实数类型的变量
    printf("Please Input a,b: ");       //输出提示信息
    scanf("%f,%f",&a,&b);              //从键盘输入两个数分别存入变量 a 和 b
    min=min1(a,b);                     //调用函数 min1(),并将返回值赋给变量 min
    printf("min=%f\n",min);            //输出结果
    return 0;
}
float min1(float x,float y)            //定义函数 min1()
{
    float z;                           //定义变量 z 是 float 实数类型
    if(x<y)                             //条件判断语句,判断 x 是否小于 y
        z=x;                           //如果 x<y,则执行该行赋值语句
    else                                //否则,即 x 不小于 y,则执行下一行赋值语句
        z=y;
    return(z);                          //返回 z 的值
}
```

运行结果为:

```
Please Input a,b: 1.2,5.4 ✓
min=1.200000
```

说明:

(1) 本程序中定义了 main()和 min1()两个函数。其中 main()函数是 C 语言程序必不可少的,min1()函数是用户根据功能需求自己定义的,称为用户自定义函数。这两个函数是相互平行的,它们可以通过调用发生联系。所有被调用的函数都必须先定义后使用,main()主函数不可以被其他函数调用。

(2) 程序的第 2 行是一条函数声明语句,当被调用函数写在主调函数的后面时,必须对被调用函数进行声明,以使系统在编译时识别。

(3) 程序的第 7 行 scanf()函数和第 9 行 printf()函数中的“%f”是格式控制符,表示输入和输出数据的类型是 float 类型,即实型,也称为单精度浮点型。

(4) 程序的第 12~第 20 行是用户自定义函数 min1(),其功能是比较两个实数的大小,并将其中较小的数返回调用该函数的语句。

1.2 C 程序的结构与书写格式

通过 1.1 节中的 3 个例子,可以看出 C 语言程序的结构和书写格式如下。

1. C 程序的结构

(1) C 程序由函数构成。一个 C 源程序可以由一个 main()主函数和若干个用户自

定义函数构成,其中 `main()` 主函数必须有而且只能有一个,通常 `main()` 函数以“`return 0;`”语句结束。函数是 C 程序的基本单位。

(2) 函数由函数首部和函数体两部分组成。如例 1-3 中的用户自定义函数 `min1()`,其中 `float min1(float x, float y)` 是函数首部,包括函数名 `min1`、函数类型 `float`、参数类型 `float` 和参数 `x` 和 `y`。注意:函数名后面的一对圆括号不可缺少。此函数中各项的含义为:

float	min1	(float	x,	float	y)
↓	↓	↓	↓	↓	↓
函数类型	函数名	参数类型	参数名	参数类型	参数名

函数体是函数首部后面一对大括号 `{}` 内的内容。函数体一般由两部分组成:对所使用的变量进行定义的说明部分和完成各种操作的执行部分。

(3) C 程序的执行总是从 `main()` 函数开始,并到 `main()` 函数结束。`main()` 函数在整个程序中的位置可以任意。

(4) C 程序的语句以分号“`;`”结束。

(5) C 程序中包含注释,以方便理解程序,注释不参与程序的执行。

2. C 程序的书写格式

为便于阅读和理解,C 程序的书写一般遵循以下规则。

(1) 一个语句占一行。用 C 语言书写程序时较为自由,既可以一行写一个语句,也可以一行写多个语句,还可以一个语句分多行来写,但为了清晰起见,建议一个语句占一行。

(2) 英文字母严格区分大小写。C 程序中英文字母严格区分大小写,一般书写 C 程序时使用小写字母。C 语言规定了 37 个有特定意义的单词,称为关键字,这些关键字在使用时必须是小写字母。

(3) 采用缩进格式的书写方法。为了看清 C 程序的层次结构,便于阅读和理解程序,C 程序一般都采用缩进格式的书写方法。缩进格式要求在书写程序时,不同结构层次的语句,从不同的起始位置开始,同一结构层次中的语句,缩进同样个数的字符位置。

(4) 为了便于阅读和理解程序,在程序中适当添加注释信息。在编写程序时应力求遵循以上书写规则,以养成良好的编程习惯。

1.3 C 语言的特点

计算机程序设计语言一般可以分为机器语言、汇编语言和高级语言。

机器语言是计算机能够直接识别的语言,其操作指令只能由二进制代码 0 和 1 构成。汇编语言是为方便记忆和编写程序,用一些助记符表示二进制代码,是一种与机器语言对应的符号化的语言,汇编语言编写的程序不能被计算机识别,必须通过专门的汇编程序将符号转换成二进制代码才能执行。高级语言是 20 世纪 50 年代发展起来使用人们习惯的自然语言编写程序的计算机语言,高级语言编写的程序计算机也不能直接识别和执行,必须通过专门的编译程序转换成机器语言才能执行,其转换的方式有两种:一种是解释方

式,即将高级语言编写的程序翻译一句执行一句;另一种是编译方式,即将高级语言编写的程序全部翻译成机器语言,生成可执行文件后再执行。C语言采用的是编译方式。

C语言是一种介于汇编语言和高级语言之间的程序设计语言,它有以下特点。

(1) 程序结构简洁紧凑。C程序由若干函数构成,各函数是相互独立的,它们通过调用发生联系,C语言是一种模块化程序设计语言。程序书写形式自由。

(2) 表达能力强且应用灵活。C语言运算符丰富,共有34种,可以组成各种类型的表达式以提高运算效率。C语言数据类型丰富,包括整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等,能实现各种数据结构的运算,从而可以适应不同的功能需求。

(3) 生成的目标程序质量好,执行效率高。C语言具有汇编语言的许多特性,允许直接访问物理地址,能进行位操作,可以直接对硬件进行操作。用C语言编写的程序经编译后生成的可执行代码比用汇编语言编写的代码执行效率仅低10%~20%。

(4) C程序可移植性好。C语言通过调用标准输入/输出库函数实现输入/输出功能,因此C语言不依赖于计算机硬件系统,从而便于在不同的计算机之间实现程序的移植。

由于C语言具有上述众多特点,已经成为程序设计的主要语言之一,被广泛应用于计算机的系统软件和应用软件的开发。

1.4 C程序的开发过程

简单的C程序从编写到得到最终结果,其开发过程如图1-1所示。

1. 编辑源程序

用高级语言编写的程序称为源程序。将C语言编写的源程序输入编辑器,并保存为文件,扩展名为.c。

2. 编译程序

编译程序就是将C源程序转换成机器语言程序,编译的作用是对源程序进行词法检查和语法检查,如果没有错误,则生成目标程序,文件扩展名是.obj;如果存在错误,则编译系统给出的出错信息分为两种,一种是错误(error),一种是警告(warning)。凡是检查出error类的错误,就不能生成目标程序,必须改正后重新编译。

编译中没有出现错误,只能说明程序中没有词法和语法错误。

3. 链接程序

使用系统的“连接程序”将目标文件与系统的库文件和系统提供的其他信息链接起来,生成可执行的二进制文件,扩展名是.exe。

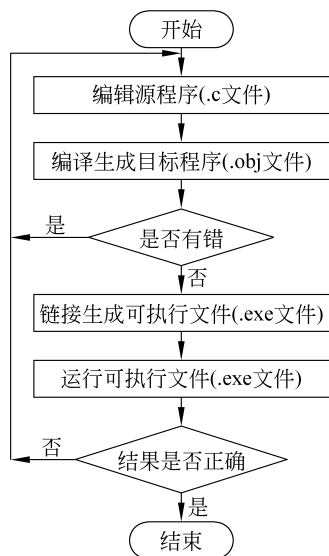


图 1-1 C程序的开发过程

4. 运行程序

运行.exe 可执行文件,得到运行结果。若运行结果不正确,则需检查并修改源程序,重复上述步骤,直到得到正确的运行结果为止。

1.5 C 程序的基本要素

1.5.1 标识符

程序中变量、类型、函数和标号等的名称统称为标识符。在 C 语言中,有两类标识符。

1. 用户自定义标识符

由用户根据需要自行定义的标识符,通常用作函数名、变量名等,如自定义函数名 Add,变量名 x 、 y 、sum 等。

标识符的命名必须遵循一定的规则。C 语言规定,标识符只能由字母(A~Z, a~z)、数字(0~9)和下划线(_)组成,并且其第一个字符必须是字母或下划线。以下标识符是合法的:

Average, x, x3, BOOK_1, sum5, _123

以下标识符是非法的,不能作为变量名或函数名:

6s(以数字开头),s * T(出现非法字符*),bowy-1(出现非法字符-)

C 语言是区分大小写的,即在标识符中,大小写字母是有区别的。例如 BOOK 和 book 是两个不同的标识符。

标识符虽然可由用户随意定义,但它是用于标识某个量的符号,因此,命名应尽量有相应的意义,以便于阅读理解,做到“见名知意”。

2. 关键字

关键字又称保留字,是系统已有的标识符,表 1-1 中列出的关键字(C99 新增 5 个关键字)对 C 编译器而言都有着特殊的意义,用户只能按其预先规定的含义来使用它们,而不能擅自改变其含义。

表 1-1 关键字

auto	break	case	char	const	continue
default	do	double	else	enum	extern
float	for	goto	if	inline ^①	int
long	register	restrict ^①	return	short	signed
sizeof	static	struct	switch	typedef	union
unsigned	void	volatile	while	_Bool ^①	_Complex ^①
_Imaginary ^①					

注: ① 仅 C99 有。

1.5.2 数据类型、常量和变量

在C语言中使用的数据都是有类型的,即通过类型说明数据的种类。数据有常量和变量之分。

1. 数据类型

C语言提供了丰富的数据类型:①系统定义的基本数据类型,包括整型、浮点型(实型)和字符型;②构造类型(由若干个基本数据类型的变量按特定的规律组合构造而成的数据),包括数组、结构体、共用体、枚举等;③指针类型;④无值类型 void。

各种数据所能表示的数据精度不同,因而它所占用的内存空间的大小也不同。在程序中直接给出的数据,计算机可以自动识别数据的类型,但当使用标识符来表示可变化的数据时,就需要考虑该数据变化的范围和精度。下面仅就基本数据类型来分析所能表示的数的精度和所占用的内存空间。表1-2列出了32位机的整型数据所占存储空间与范围。

表 1-2 32 位机的整型数据

数据类型	字节数	数值范围
整型 int	4	$-2147483648 \sim 2147483647 (-2^{31} \sim 2^{31} - 1)$
无符号整型 unsigned int	4	$0 \sim 4294967295 (0 \sim 2^{32} - 1)$
短整型 short int	2	$-32768 \sim 32767 (-2^{15} \sim 2^{15} - 1)$
长整型 long int	4	$-2147483648 \sim 2147483647 (-2^{31} \sim 2^{31} - 1)$
无符号长整型 unsigned long int	4	$0 \sim 4294967295 (0 \sim 2^{32} - 1)$

另外,C99标准提供了两个额外的数据类型:long long int 和 unsigned long long int 以满足日益增长的对超大型整数的需求。同时为了支持64位运算的新处理器的能力,这两个long long类型要求至少64位宽。long long int 的取值范围为: $-9223372036854775808 \sim 9223372036854775807 (-2^{63} \sim 2^{63} - 1)$,而 unsigned long long int 的取值范围为: $0 \sim 18446744073709551615 (0 \sim 2^{64} - 1)$ 。

C标准没有说明float、double 和 long double 类型提供的精度到底是多少,因为不同的计算机可以用不同的方法存储浮点数。大多数现代计算机都遵循IEEE754标准的规范。表1-3列出了根据IEEE标准实现时浮点类型数据所占存储空间、范围及精度。

表 1-3 浮点类型(IEEE 标准)

数据类型	字节数	有效位数	数值范围(绝对值)
单精度浮点型 float	4	6~7	$1.17549 \times 10^{-38} \sim 3.40282 \times 10^{38}$
双精度浮点型 double	8	15~16	$2.22507 \times 10^{-308} \sim 1.79769 \times 10^{308}$
扩展精度浮点型 long double	10(16)	18~19	约 $10^{-4932} \sim 10^{4932}$

说明: long double 类型的长度随着机器的不同而变化,而最常见的大小是80位和128位。

C语言设置字符类型char的目的是存储字母和标点符号之类的字符。字符类型所

占存储空间为 1 字节。C 语言把字符当作小整数进行处理,因此字符类型 char 存储的是整数而不是字符。char 类型可以是 signed char 类型(取值范围为 $-128 \sim 127$),也可以是 unsigned char 类型(取值范围为 $0 \sim 255$),具体取决于编译器。也就是说,不同机器上的 char 类型可能有不同范围的取值,因此为了使程序保持良好的可移植性,我们所声明的 char 类型变量的值应该限制在 signed char 与 unsigned char 的交集范围内。

2. 常量

常量是在程序执行过程中其值不能被改变的量。常量有各种不同的数据类型,常量的类型通常由书写格式决定。例如:25 是整型常量,25.36 是浮点型常量,'A'是字符常量,"ABC"是字符串常量。

1) 整型常量的表示形式

计算机中的数据都以二进制形式存储。在 C 语言中,使用的整型常量有十进制、八进制和十六进制 3 种。

(1) 十进制整数:由数字 $0 \sim 9$ 组成,没有前缀,不能以 0 开始。数字前可带正负号。如 221、-128、0、+9。

(2) 八进制整数:由数字 $0 \sim 7$ 组成,以 0 为前缀。如 015。

(3) 十六进制整数:由数字 $0 \sim 9$ 和字母 A~F(a~f)组成,以 0X 或 0x 为前缀。如 0X2A。

在程序中是根据前缀来区分各种进制数的,因此在书写常数时不要把前缀弄错,造成结果不正确。

整型常量的后缀:若整数后面加 L 或 l 表示是长整型数,如 158L、012L、0x15L 等。若整数后面加 U 或 u 表示是无符号整型数,如 358u、0x3A8u 等。

2) 浮点型常量的表示形式

计算机中的实数也称浮点数。如 3.14159、-42.8 等都是浮点型常量。在 C 语言中,浮点型常量的表示方法有两种。

(1) 十进制小数形式:由数字 $0 \sim 9$ 和小数点组成。注意必须有小数点,如 0.123、-12.36、.78、18.等都是合法的表示形式,其中,.78 等效于 0.78,18. 等效于 18.0。如果没有小数点,则不是浮点型常量。

(2) 指数形式:在实际应用中会遇到绝对值很大或很小的数,这时将其写成指数形式,更显得直观方便,如 0.00000234 写成 2.34×10^{-6} ,或者 0.234×10^{-7} ,但程序编辑时不能输入上标,因此 C 语言中用字母 e 或 E 来代表以 10 为底的指数。其一般形式为:

$a \text{ e } n$ (a 为十进制数,n 为十进制整数)

其值为 $a \times 10^n$ 。

指数形式表示时要求 e 的前后均有数字,且 e 后面为整数。如 1.1e5,3.6e-2,0.6e7,-2.7e-2 等都是合法的表示形式,而 E7、2.7E、.e3 等都是不合法的表示形式。

一个浮点数可以有多种指数表示形式。例如 123.456 可以表示为 123.456e0,12.3456e1,1.23456e2,0.123456e3,0.0123456e4 等。

浮点型常量的后缀：浮点型常量的默认类型是 double 类型。若浮点常量加后缀 f 或 F, 表示该常量是 float 类型, 如 35.6f、12.5F 等; 若浮点常量加后缀 l 或 L, 表示该常量是 long double 类型。

3) 字符常量

字符常量是用单引号括起来的一个字符。

例如: 'a'、'A'、'*'、'\$'、'?' 都是合法字符常量。字符常量只能用单引号括起来, 不能用双引号或其他括号; 字符常量只能是单个字符; 字符可以是字符集中任意字符。

转义字符是一种特殊的字符常量, 它是以反斜线 \ 开头, 后跟一个或几个字符。转义字符具有特定的含义, 不同于字符原有的意义, 故称“转义”字符。常用的转义字符如表 1-4 所示。

表 1-4 常用的转义字符及其含义

转义字符	含 义	ASCII 码(十进制数)
\n	换行, 将当前位置移到下一行开头	10
\t	水平制表(跳到下一个 Tab 位置)	9
\b	退格, 将当前位置移到前一列	8
\r	回车, 将当前位置移到本行开头	13
\f	换页, 将当前位置移到下页开头	12
\a	发出铃声	7
\\	代表一个反斜杠字符“\”	92
\'	代表一个单引号	39
\"	代表一个双引号	34
\ddd	1~3 位八进制数所代表的字符	
\xhh	1~2 位十六进制数所代表的字符	

转义字符主要用来表示那些用一般字符不便于表示的控制代码。表中的 \ddd 和 \xhh 正是为此而提出的。ddd 和 hh 分别为八进制和十六进制的 ASCII 码。如 \101 表示字母 'A', \102 表示字母 'B', \134 表示反斜线, \X0A 表示换行等。

4) 字符串常量

字符串常量是由一对双引号括起的字符序列。例如: "hello"、"C program"、"\$ 12.36" 等都是字符串。无论双引号内是否包含字符、包含多少个字符, 都代表一个字符串常量。

字符串常量和字符常量是不同的量, 要特别注意它们之间的区别。

(1) 字符常量由单引号括起来, 字符串常量由双引号括起来。

(2) 字符常量只能是单个字符, 字符串常量则可以含零个或多个字符。

(3) 字符常量占 1B 的内存空间。字符串常量占的内存字节数等于字符串中字符个数加 1。增加的一个字节存放字符 \0 (ASCII 码为 0)。这是字符串结束的标志。

例如: 如果有一个字符串常量 "hello", 实际上在内存中是

h	e	l	l	o	\0
---	---	---	---	---	----

它占内存单元不是 5 字节,而是 6 字节,最后一个字符为'\0'。但在输出时不输出'\0'。

字符常量'a'和字符串常量"a"虽然都只有一个字符,但在内存中的情况是不同的。

'a'在内存中占 1 字节,可表示为:



"a"在内存中占 2 字节,可表示为:



特别注意,在写字符串常量时不必加'\0',因为'\0'是系统自动加上的。

5) 符号常量

在 C 语言中,还可以用一个标识符来表示一个常量,称为符号常量。符号常量在使用之前必须先定义,其一般形式为:

```
#define 标识符 字符串
```

其中,#define 是一条预处理命令(预处理命令都以#开头),称为宏替换,其功能是在程序中使用到“标识符”时都用其后的“字符串”替换。

符号常量在程序中不能被改变,也不能被赋值。符号常量通常用大写字母表示。使用符号常量可以增加程序的可读性,而且能做到“一改全改”,增强程序的可维护性。

【例 1-4】 符号常量的使用。

```
#define PI 3.14159           //定义符号常量
#include<stdio.h>
int main(void)
{
    float r,v;
    r=4;
    v=4.0/3*PI*r*r*r;        //计算圆球体积
    printf("v=%.2f\n",v);
    return 0;
}
```

运行结果为:

```
v=268.08
```

程序中用标识符 PI 代表,用#define 命令行定义 PI 代表字符串 3.14159,在程序编译过程中所有 PI 均用 3.14159 代替。

6) const 常量

const 是 C 语言的关键字,用 const 定义的标识符在程序运行过程中不允许被改动,这在一定程度上提高了程序的安全性和可读性。const 的语法格式如下: