

第 1 章 C 语言概述



本章主要对 C 语言的一些基础知识及其字符集和词汇进行介绍,使读者在学习 C 语言之前对其有个全面的认识。



- 程序与程序设计语言简介
- C 源程序的结构特点
- C 语言的字符集
- C 语言词汇

任务 1 程序与程序设计语言简介



在学习 C 语言之前,先来了解一下程序与程序设计语言的基础知识。

阶段 1 程序简介

程序(program)是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合。程序是由序列组成的,用于告诉计算机如何完成一个具体的任务。

程序可以看做为实现预期目的而进行操作的一系列语句和指令,一般分为系统程序和应用程序两大类。系统程序就是为使计算机执行一个或多个操作,或执行某一任务,按序设计的计算机指令的集合,由程序计数器(Program Counter)控制。应用程序是用汇编语言、高级语言等开发语言编制出来的可以运行的文件,在计算机中称为可执行文件(后缀名一般为.exe)。我们玩的游戏就是应用程序(现在后缀名多为.swf 的 Flash 影片类的游戏也比较

流行)。

阶段 2 程序设计语言

程序设计语言,通常简称为编程语言,是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧,用来向计算机发出指令。一种计算机语言让程序员能够准确地定义计算机所需要使用的数据,并精确地定义在不同情况下所应当采取的行动。

在过去的几十年间,大量的程序设计语言被发明、被取代、被修改或被组合在一起。尽管人们多次试图创造一种通用的程序设计语言,却没有一次尝试是成功的。之所以有那么多种不同的编程语言存在,很重要的原因是编写程序的初衷各不相同;有许多用于特殊用途的语言,只在特殊情况下使用。例如,PHP 专门用来显示网页;Perl 更适合文本处理;C 语言被广泛用于操作系统和编译器(所谓的系统编程)的开发。

高级程序设计语言(也称高级语言)的出现使得计算机程序设计语言不再过度地依赖某种特定的机器或环境。这是因为高级语言在不同的平台上会被编译成不同的机器语言,而不是直接被机器执行。最早出现的编程语言之一 FORTRAN 的一个主要目标,就是为了实现平台独立。

虽然大多数的语言既可被编译(compiled)又可被解译(interpreted),但大多数只在一种情况下能够良好运行。在一些编程系统中,程序要经过几个阶段的编译,一般而言,后阶段的编译往往更接近机器语言。这种常用的使用技巧最早在 1960 年代末用于 BCPL:编译程序先编译一个叫做“0 代码”的转换程序(representation),然后再使用虚拟机转换到可以运行于机器上的真实代码。这种成功的技巧之后又用于 Pascal 和 P-code,以及 Smalltalk 和二进制码,虽然在很多时候,中间过渡的代码往往是解译,而不是编译的。

如果所使用的翻译的机制是将所要翻译的程序代码作为一个整体翻译,之后运行内部格式,那么这个翻译过程就称为编译。因此,一个编译器是一个将可阅读的程序文本(叫做源代码)作为输入的数据,然后输出可执行文件(object code)。所输出的可执行文件可以是机器语言,由计算机的中央处理器直接运行,或者是某种模拟器的二进制代码。如果程序代码是在运行时才即时翻译,那么这种翻译机制就被称为解译。经解译的程序运行速度往往比编译的程序慢,但往往更具灵活性,因为它们能够与执行环境互相作用。

任务2 C源程序的结构特点



本任务主要讲解C源程序的结构特点。

阶段1 C语言简介

C语言是在20世纪70年代初问世的。1978年美国电话电报公司(AT&T)贝尔实验室正式发表了C语言。同时由B.W.Kernighan和D.M.Ritchie合著了著名的“THE C PROGRAMMING LANGUAGE”一书。通常简称为“K&R”,也有人称之为“K&R”标准。但是,在“K&R”中并没有定义一个完整的标准C语言。后来由美国国家标准学会在此基础上制定了一个C语言标准,于1983年发表,通常称之为ANSI C。

早期的C语言主要是用于UNIX系统。由于C语言的强大功能和各方面的优点逐渐为人们所认识,到了20世纪80年代,开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到广泛的使用。C语言成为当代最优秀的程序设计语言之一。

C语言是一种结构化语言。它层次清晰,便于按模块化方式组织程序,易于调试和维护。C语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型,便于实现各类复杂的数据结构,还可以直接访问内存的物理地址,进行位(bit)一级的操作。由于C语言实现了对硬件的编程操作,因此它集高级语言和低级语言的功能于一体,既可用于系统软件的开发,也适合于应用软件的开发。此外,C语言还具有效率高、可移植性强等特点。因此,C语言被广泛地移植到了各种类型的计算机上,从而形成了多种版本的C语言。

目前最流行的C语言有以下几种:

- Microsoft C 或称 MS C
- Borland Turbo C 或称 Turbo C
- AT&T C

这些C语言版本不仅实现了ANSI C标准,而且在此基础上各自作了一些扩充,使之更加方便、完美。

阶段2 C源程序的结构特点

- 一个C语言源程序可以由一个或多个源文件组成。
- 每个源文件可由一个或多个函数组成。
- 一个源程序不论由多少个文件组成,都有且只有一个main函数,即主函数。

- 源程序中可以有预处理命令(include 命令仅为其中的一种),预处理命令通常应放在源文件或源程序的最前面。

- 每一个说明、每一个语句都必须以分号结尾。但预处理命令、函数头和花括号“}”之后不能加分号。

- 标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符,也可不再加空格来间隔。

为了进一步说明 C 语言源程序结构的特点,先看以下几个程序。这几个程序由简到难,表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍,但可从这些例子中了解到一个 C 源程序的基本组成部分和书写格式。

例 1.1

```
main()
{
printf("您好! 欢迎学习 C 语言\n");
}
```

main 是主函数的函数名,表示这是一个主函数。每一个 C 源程序都必须有,且只能有一个主函数(main 函数)。函数调用语句,printf 函数的功能是把要输出的内容送到显示器去显示。printf 函数是一个由系统定义的标准函数,可在程序中直接调用。

例 1.2

```
#include <math.h> /* include 称为文件包含命令。它包含扩展名为.h 的文件,也称为头文件或首部文件 */
#include <stdio.h>
main()
{
double x,s; /* 定义两个实数变量,以被后面程序使用 */
printf("input number:\n"); /* 显示提示信息 */
scanf("% lf",&x); /* 从键盘获得一个实数 x */
s = sin(x); /* 求 x 的正弦,并把它赋给变量 s */
printf("sine of % lf is % lf\n",x,s); /* 显示程序运算结果 */
} /* main 函数结束 */
```

上述程序的功能是从键盘输入一个数 x,求 x 的正弦值,然后输出结果。在 main()之前的两行称为预处理命令(详见后面)。预处理命令还有其他几种,这里的 include 称为文件包含命令,其意义是把尖括号(<>)或引号("")内指定的文件包含到本程序来,成为本程序的一部分。被包含的文件通常是由系统提供的,其扩展名为.h。这个文件也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型。因此,凡是在程序中调用

一个库函数时,都必须包含该函数原型所在的头文件。在本例中,使用了三个库函数:输入函数 `scanf`, 正弦函数 `sin`, 输出函数 `printf`。`sin` 函数是数学函数,其头文件为 `math.h` 文件,因此在程序的主函数前用 `include` 命令包含了 `math.h`。`scanf` 和 `printf` 是标准输入输出函数,其头文件为 `stdio.h`,因此在主函数前也用 `include` 命令包含了 `stdio.h` 文件。

需要说明的是,C语言规定对 `scanf` 和 `printf` 这两个函数可以省去对其头文件的包含命令。所以在本例中也可以删去第 2 行的包含命令 `#include`。同样,在例 1.1 中使用了 `printf` 函数,也省略了包含命令。

本例中的主函数体中又分为两部分,一部分为说明部分,另一部分为执行部分。说明是指变量的类型说明。例 1.2 中未使用任何变量,因此无说明部分。C语言规定,源程序中所有用到的变量都必须先说明后使用,否则将会出错。这一点是编译型高级程序设计语言的一个特点,与解释型的 BASIC 语言是不同的。说明部分是 C 源程序结构中很重要的组成部分。例 1.2 中使用了两个变量 `x,s`,用来表示输入的自变量和 `sin` 函数值。由于 `sin` 函数要求这两个量必须是双精度浮点型,故用类型说明符 `double` 来说明这两个变量。说明部分后的四行为执行部分或称为执行语句部分,用以完成程序的功能。执行部分的第 1 行是输出语句,调用 `printf` 函数在显示器上输出提示字符串,请操作人员输入自变量 `x` 的值。第 2 行为输入语句,调用 `scanf` 函数,接受键盘上输入的数并存入变量 `x` 中。第 3 行是调用 `sin` 函数并把函数值送到变量 `s` 中。第 4 行是用 `printf` 函数输出变量 `s` 的值,即 `x` 的正弦值。程序结束。

例 1.3

```
printf("input number:\n");
scanf("% lf", &x);
s = sin(x);
printf("sine of % lf is % lf\n", x, s);
```

运行本程序段时,首先在显示器屏幕上给出提示串 `input number:`,这是由执行部分的第一行完成的。用户在提示下从键盘上键入某一数,如 5,按下回车键,接着在屏幕上给出计算结果。

2. 输入和输出函数

在例 1.1 和例 1.2 中用到的输入和输出函数 `scanf` 和 `printf`,将在第 3 章中作详细介绍。这里先简单介绍一下它们的格式,以便下面使用。`scanf` 和 `printf` 这两个函数分别称为格式输入函数和格式输出函数。其意义是按指定的格式输入输出值。因此,这两个函数在括号中的参数表都由以下两部分组成:格式控制串及参数表。格式控制串是一个字符串,必须用双引号括起来,它表示了输入输出量的数据类型。各种类型的格式表示法可参阅第 3 章。在 `printf` 函数中,还可以在格式控制串内出现非格式控制字符,这时在屏幕上将原文显示。参数表中给出了输入或输出的量。当有多个量时,用逗号间隔。

例如:

```
printf("sine of % lf is % lf\n",x,s);
```

其中,%lf为格式字符,表示按双精度浮点数处理。它在格式串中两次出现,对应了x和s两个变量。其余字符为非格式字符,照原样输出在屏幕上。

例 1.4

```
int max(int a,int b);           /* 函数说明 */
main(){                         /* 主函数 */
int x,y,z; /* 变量说明 */
printf("input two numbers:\n");
scanf("% d % d",&x,&y);

                               /* 输入 x,y 值 */
z = max(x,y);                  /* 调用 max 函数 */
printf("maxmum = % d",z);      /* 输出 */
}
int max(int a,int b){          /* 定义 max 函数 */
if(a>b)return a;else return b; /* 把结果返回主调函数 */
}
```

例 1.4 中程序的功能是由用户输入两个整数,程序执行后输出其中较大的数。本程序由两个函数组成,主函数和 max 函数。函数之间是并列关系。可从主函数中调用其他函数。max 函数的功能是比较两个数,然后把较大的数返回给主函数。max 函数是一个用户自定义函数,因此在主函数中要给出说明(程序第 3 行)。可见,在程序的说明部分中,不仅可以有变量说明,还可以有函数说明。关于函数的详细内容将在第 5 章介绍。在程序的每行后用“/*”和“*/”括起来的内容为注释部分,程序不执行注释部分。

例 1.4 中程序的执行过程是,首先在屏幕上显示提示串,请用户输入两个数,回车后由 scanf 函数语句接收这两个数送入变量 x,y 中,然后调用 max 函数,并把 x,y 的值传送给 max 函数的参数 a,b。在 max 函数中比较 a,b 的大小,把大者返回给主函数的变量 z,最后在屏幕上输出 z 的值。

3. 书写程序时应遵循的规则

从书写清晰,便于阅读、理解、维护的角度出发,在书写程序时应遵循以下规则:

(1)一个说明或一个语句占一行。

(2)用{}括起来的部分,通常表示了程序的某一层结构。“{”和“}”一般与该结构语句的第一个字母对齐,并单独占一行。

(3)低一层次的语句或说明可比高一层次的缩进若干格后书写。以便看起来更加清晰,增加程序的可读性。

任务3 C语言的字符集



字符是组成语言的最基本的元素。C语言字符集由字母、数字、空白符、标点和特殊字符组成。在字符常量、字符串常量和注释中,可以使用汉字或其他可表示的图形符号。本任务将简要介绍这方面的内容。

1. 字母

小写字母 a~z 共 26 个,大写字母 A~Z 共 26 个。

2. 数字

0~9 共 10 个。

3. 空白符

空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其他地方出现时,只起间隔作用,编译程序对它们忽略。因此在程序中使用空白符与否,对程序的编译不产生影响;但在程序中适当的地方使用空白符,将增加程序的清晰性和可读性。

4. 标点和特殊字符

包括常用的标点和一些特殊字符,和其他文档中的相同。

任务4 C语言词汇



在C语言中使用的词汇分为6类:标识符、关键字、运算符、分隔符、常量、注释符等。本任务将分别进行介绍。

阶段1 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外,其余都由用户自定义。C语言规定,标识符只能是字母(A~Z, a~z)、数字(0~9)、下划

线(-)组成的字符串,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:

a, x, 3x, BOOK 1, sum5

以下标识符是非法的:

3s 以数字开头

s * T 出现非法字符 *

-3x 以减号开头

bowy-1 出现非法字符-(减号)

在使用标识符时还必须注意以下几点:

(1)标准 C 语言不限制标识符的长度,但它受各种版本的 C 语言编译系统限制,同时也受到具体机器的限制。例如,在某版本 C 语言中规定标识符前 8 位有效,当两个标识符前 8 位相同时,则被认为是同一个标识符。

(2)在标识符中,大小写是有区别的。例如,BOOK 和 book 是两个不同的标识符。

(3)标识符虽然可由程序员随意定义,但标识符是用于标识某个量的符号。因此,命名应尽量有相应的意义,以便阅读理解,做到“顾名思义”。

阶段 2 关键字

关键字是由 C 语言规定的具有特定意义的字符串,通常也称为保留字。用户定义的标识符不应与关键字相同。C 语言的关键字分为以下几类。

1. 类型说明符

用于定义或说明变量、函数和其他数据结构的类型。如前面例题中用到的 int, double 等。

2. 语句定义符

用于表示一个语句的功能。如例 1.4 中用到的 if-else 就是条件语句的语句定义符。

3. 预处理命令字

用于表示一个预处理命令。如前面各例中用到的 include。

阶段 3 运算符

C 语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式,表示各种运算功能。运算符由一个或多个字符组成。这部分将在第 3 章详细介绍。

阶段 4 分隔符

在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中,分隔各个变量。空格多用于语句各单词之间,作间隔符。在关键字、标识符之间必须有一个以上的空格符作间隔,否则将会出现语法错误。例如,把 int a;写成 inta;,C 编译器会

把 `inta` 当成一个标识符处理,其结果必然出错。

阶段5 常量

C语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。在第3章中将专门进行介绍。

阶段6 注释符

C语言的注释符是以“`/*`”开头并以“`*/`”结尾的串。在“`/*`”和“`*/`”之间的即为注释。程序编译时,不对注释作任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来,使翻译跳过不作处理,待调试结束后再去掉注释符。

本章小结

本章主要介绍了C语言的基础知识,包括程序与程序设计语言简介、C源程序的结构特点、C语言的字符集和C语言词汇,为学习C语言做好准备。

本章习题

1. 填空题

- (1) C语言的字符集包括_____、_____、_____和_____。
- (2) C语言是通过_____来进行输入和输出的。

2. 选择题

- (1) 以下不是C语言的特点的是()。
- A. 语言简洁紧凑
B. 能够编制出功能复杂的程序
C. 可以直接对硬件操作
D. 移植性好
- (2) 下列字符序列中,不可用作C语言标识符的是()。
- A. `abc123` B. `no.1` C. `_123_` D. `-_ok`
- (3) 下列字符列中,正确的C语言标识符是()。
- A. `_buy_2` B. `2_buy` C. `?_buy` D. `buy?`
- (4) 下列符号中,不属于转义字符的是()。
- A. `\\` B. `\0xAA` C. `\t` D. `\0`
- (5) 下列不属于C语言关键字的是()。

A. int B. break C. while D. character

(6)下列属于 C 语言提供的合法关键字的是()。

A. Float B. signed C. integer D. Char

(7)以下不能定义为用户标示符的是()。

A. scanf B. Void C. _3com_ D. int

3.问答题

(1)简述 C 语言的结构特点。

(2)书写程序时需要注意什么?