



第1章 C语言程序设计概述

本章目录

- 1.1 程序设计基础
- 1.2 C语言的发展与特点
- 1.3 C语言的字符集与词汇
- 1.4 在VC++6.0环境编译运行C语言程序

教学目标

- 掌握C语言标识符的命名方法
- 熟悉C语言源程序的基本结构与书写风格
- 了解C语言程序的结构，**main**函数和其他函数
- 了解头文件，数据说明，函数的开始和结束标志以及程序中的注释

1.1 程序设计基础

在基于最常见的冯·诺依曼体系结构(又称Harvard Architecture)的计算机上, 计算机是在程序的控制下完成用户指定的各项工作的, 基于这种体系的计算机如果没有程序的支持将无法工作。要让计算机能够正常的运行, 需要预先编写计算机工作步骤的指令序列, 即系统程序; 要利用计算机来解决一个具体的实际问题, 同样需要编写程序, 即应用程序。这些都要进行程序设计工作。

1.1 程序设计基础

1.1.1 程序与程序设计

所谓程序，是指为了得到某种结果而可以由计算机等具有信息处理能力的装置执行的代码化指令序列，或者可以被自动转换成代码化指令序列的符号化指令序列或者符号化语句序列。就是一系列遵循一定规则和思想并能正确完成指定工作的代码（也称为指令的序列）。通常一个计算机程序主要描述两部分内容，一是问题的每个对象及他们之间的关系，即数据结构的内容；二是对这些对象进行处理的动作和动作的先后顺序，即求解的算法。因此，程序也可以用经典的公式来表示：

程序 = 数据结构 + 算法

1.1 程序设计基础

一个设计合理的数据结构往往可以简化算法，而且一个好的程序有可靠性、易读性、可维护性等良好特性。

程序设计是给出解决特定问题程序的过程，是软件构造活动中的重要组成部分。就是根据计算机要完成的任务，提出相应的需求，在此基础上设计数据结构和算法，然后用某种程序设计语言编写相应的程序代码。程序设计往往以某种程序设计语言为工具，给出这种语言下的程序。程序设计过程包括分析、设计、编码、测试、排错等不同阶段。

1.1 程序设计基础

1.1.2 程序设计语言

程序设计语言用来书写计算机程序的工具。语言的基础是一组记号和一组规则。根据规则由记号构成的记号串的总体就是语言。在程序设计语言中，这些记号串就是程序。程序设计语言有3个方面的因素，即语法、语义和语用。语法表示程序的结构或形式，即表示构成语言的各个记号之间的组合规律。语义表示程序的含义，即表示按照各种方法所表示的各个记号的特定含义。程序设计的语言种类繁多，但最常用的不过十多种。程序设计语言按照其发展的过程，可以分为机器语言、汇编语言和高级语言三类。

1.1 程序设计基础

机器语言是由0和1这样的二进制代码按一定规则组成的、能被机器直接理解和执行的指令集合。用机器语言编写的程序可以被计算机直接识别和执行，执行效率高。但由于不同的机器的指令系统各不相同，从而导致机器语言程序通用性差。另外，由于机器语言存在指令不直观，难以辨认和记忆等诸多缺点，因此，很少有人用机器语言直接编程。

1.1 程序设计基础

汇编语言，又称为符号语言。为了便于理解和记忆，人们采用能够帮助记忆的指令助记符来代替机器语言指令代码中的操作码，用地址符号或十进制数来代替操作数。指令助记符一般采用描述该指令功能的英文单词的缩写，如用**ADD**表示加法操作、**SUB**表示减法操作、**JMP**表示程序跳转等等。这种指令助记符的语言就是汇编语言。计算机不能直接执行用汇编语言编写的程序，必须由一种专门的翻译程序（即汇编程序）将汇编语言源程序翻译成机器语言程序，计算机才能执行。

1.1 程序设计基础

机器语言和汇编语言都是面向机器的语言，一般称为低级语言。它们对机器的依赖性很大，用它们开发出的程序通用性差，而且要求程序的开发者必须熟悉和了解计算机硬件的每一个细节。随着计算机技术的发展及计算机应用领域的不断扩大，计算机用户的队伍也在不断壮大，从20世纪50年代中期开始，逐步发展了面向问题的程序设计语言，称为高级语言。高级语言与具体的计算机硬件无关，其表达方式接近于被描述的问题，接近于自然语言和数学语言，易为人们接受和掌握。用高级语言编写程序要比用低级语言容易得多，大大简化了程序的编制和调试过程，使编程效率得到大幅度的提高。

用高级语言编写的程序易学、易读、易修改，通用性好。但是，计算机同样不能直接执行高级语言程序，必须经过语言处理程序的翻译后才能被机器接受。

1.1 程序设计基础

1.1.3 语言处理程序

计算机可以直接识别和执行的只有机器语言程序。用汇编语言和高级语言编写的程序，都必须经过一个翻译过程才能转换为计算机所能识别的机器语言程序，实现这个翻译过程的工具是语言处理程序，即翻译程序。翻译程序除了要完成语言间的转换外，还要进行语法、语义等方面的检查。不同的程序设计语言编写出的程序，有各自不同的翻译程序。翻译程序可分为汇编程序和高级语言翻译程序；高级语言翻译程序有两种工作方式：解释方式和编译方式，对应的是解释程序和编译程序。

1.1 程序设计基础

1. 汇编程序

汇编程序是一种由专业的软件开发商提供的系统软件，它能将用汇编语言编写的源程序翻译成某种类型的计算机的机器语言程序，即目标程序，这一翻译过程称为汇编。图1.1是汇编程序的功能示意图。



图 1.1 汇编过程示意图

1.1 程序设计基础

2. 解释程序

解释程序的工作过程是将高级语言源程序逐句地读入，每读入一个语句都要对它进行分析和解释，若有错误就即时中断其解释过程，并通知用户进行修改，若没有错误就按照解释结果执行所要求的操作。其工作过程如图1.2所示。



图 1.2 解释过程示意图

1.1 程序设计基础

BASIC、LISP等语言采用解释方式。解释方式灵活、方便，交互性好，解释执行程序的过程中也不会产生目标程序。但因为是边解释边执行，所以程序的执行速度很慢，如果源程序中出现循环结构，解释程序也要重复多次地解释循环体中的每一条语句，造成很大浪费，而且解释方式在运行源程序时离不开翻译程序。

1.1 程序设计基础

3. 编译程序

编译程序的功能是将用高级语言编写的源程序翻译成机器语言程序，即目标程序，这一翻译过程称为编译。但目标程序还不能立即装入机器执行，因为还没有连接成一个整体，在目标程序中还可能要调用一些其他语言编写的程序和标准程序库中的标准子程序，所有这些程序通过连接程序将目标程序和有关的程序库组合成一个完整的可执行程序。产生的可执行程序可以脱离编译程序和源程序独立存在并反复使用。故编译方式执行速度快，但每次修改源程序后，都必须重新编译。其工作过程如图1.3所示。

1.1 程序设计基础

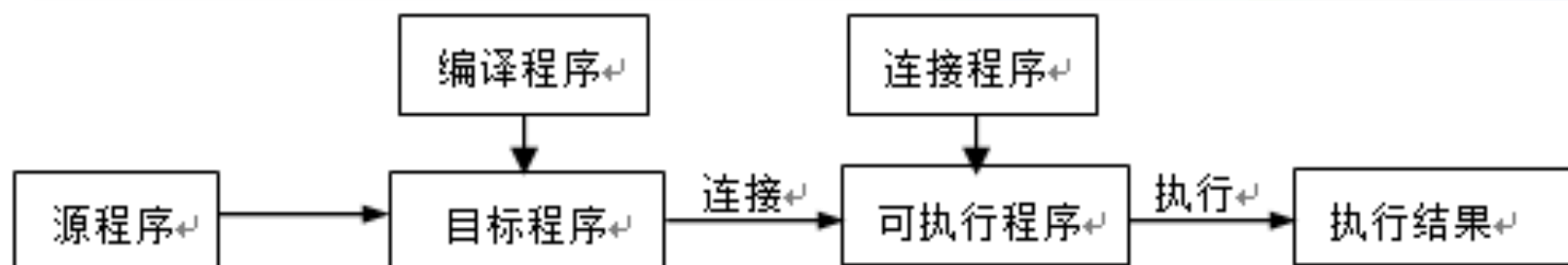


图 1.3 编译过程示意图

与汇编程序不同的是，编译程序往往需要将一条高级语言的语句转换成若干条机器语言指令，而且翻译的过程也要复杂得多。一般高级语言（C/C++、PASCAL、FORTRAN、COBOL等）都是采用编译方式。

1.1 程序设计基础

要设计出一个好的程序，必须了解利用计算机解决实际问题的过程，掌握程序设计的基本技术，熟练地掌握一种程序设计语言。计算机解决问题的基本过程如图1.4所示。

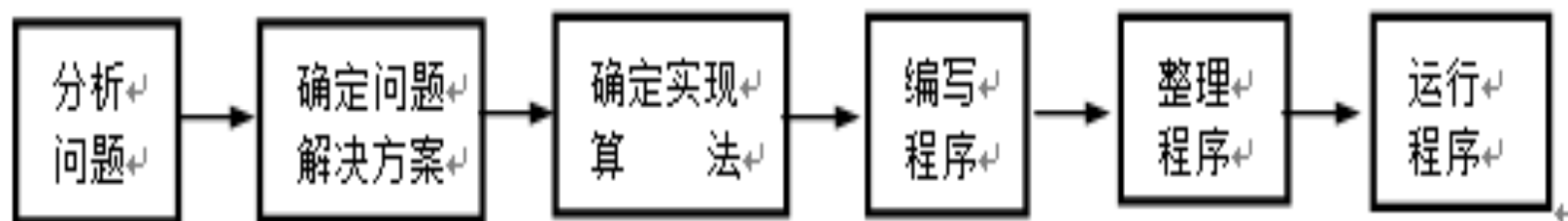


图1.4 计算机解决问题的基本过程

1.1 程序设计基础

如何才能编写出高质量的程序呢？下面是设计程序时应遵循的基本原则：

1. 正确性。正确性是指程序本身必须具备且只能具备程序设计规格说明书中所列举的全部功能。它是判断程序质量的首要标准。

2. 可靠性。可靠性是指程序在多次反复使用过程中不失败的概率。

3. 简明性。简明性的目标是要求程序简明易读。

4. 有效性。程序在计算机上运行需要使用一定数量的计算机资源，如CPU的时间、存储器的存储空间。有效性就是要在一定的软、硬件条件下，反映出程序的综合效率。

1.1 程序设计基础

5. 可维护性。程序的维护可分为校正性维护、适应性维护和完善性维护。一个软件的可维护性直接关系到程序的可用性，因此应特别予以关注。

6. 可移植性。程序主要与其所完成任务有关，但也与它的运行环境有着一定的联系。软件的开发应尽可能远离机器的特征，以提高它的可移植程度。

为了有效地进行程序设计，不仅要掌握一门高级语言，还应该学会对各类问题拟定出有效的解题步骤——即算法设计。有了正确的算法，才能够编制程序。算法的好坏，决定了程序的优劣，因此，程序设计的核心任务之一就是设计算法。

1.2 C语言的发展与特点

1.2.1 C语言的产生与发展

C语言是1972年由美国的Dennis Ritchie设计发明的，并首次在UNIX操作系统的DEC PDP-11计算机上使用。它由早期的编程语言BCPL (Basic Combind Programming Language) 发展演变而来。在1970年，AT&T贝尔实验室的Ken Thompson根据BCPL语言设计出较先进的并取名为B的语言，最后导致了C语言的问世。1978年由美国电话电报公司(AT&T)贝尔实验室正式发表了C语言。同时由B.W.Kernighan和D.M.Ritchit合著了著名的“The C Programming Language”一书。

1.2 C语言的发展与特点

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认识，到了八十年代，C开始进入其它操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用，成为当代最优秀的程序设计语言之一。

随着微型计算机的日益普及，出现了许多C语言版本。由于没有统一的标准，使得这些C语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准研究所(ANSI)为C语言制定了一套ANSI标准，成为现行的C语言标准。

1.2 C语言的发展与特点

1.2.2 C语言的特点

C语言是目前国际上流行的一种结构化的程序设计语言，它不仅是开发系统软件的很好工具，而且也是很好的应用软件开发程序设计语言。因此，它深受广大程序设计者的欢迎。C语言之所以能被推广并广泛应用，概括地说主要有如下特点：

1. C语言简洁、紧凑，使用方便、灵活。ANSI C一共只有32个关键字（如表1所示）和9种控制语句（参见表1-1），程序书写自由，主要用小写字母表示，压缩了一切不必要的成分。
2. 运算符丰富。共有34种。C把括号、赋值、逗号等都作为运算符处理。从而使C的运算类型极为丰富，可以实现其他高级语言难以实现的运算。
3. 数据结构类型丰富。
4. 具有结构化的控制语句。

1.2 C语言的发展与特点

5. 语法限制不太严格，程序设计自由度大。

6. C语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此有人把它称为中级语言。

7. 生成目标代码质量高，程序执行效率高。

8. 与汇编语言相比，用C语言写的程序可移植性好。

但是，C语言对程序员要求也高，程序员用C写程序会感到限制少、灵活性大，功能强，但较其他高级语言在学习上要困难一些。

下面通过介绍几个简单的C程序，使读者对C程序有一个大概的了解。这些例子虽然简单，但反映了一般C程序的特点以及基本组成。

1.2 C语言的发展与特点

【例1.1】 编写一个C程序，其运行结果是在显示器屏幕的当前光标位置处显示如下字符串：“世界，您好！”。

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

```
main()
```

```
{
```

```
    printf(“世界，您好！ \n”);
```

```
}
```


1.2 C语言的发展与特点

`main`是主函数的函数名，每一个C源程序都必须有且只有一个主函数(`main`函数)；函数体由花括号`{ }`括起来。

`printf()` 函数是由系统提供的标准库函数，可在程序中直接调用，其功能是把要输出的内容送到显示器去显示。双引号内的字符串“世界，您好！”原样输出。

“`\n`”是换行符，即在输出“世界，您好！”后回车换行。

语句最后的分号“`;`”不能省略。

1.2 C语言的发展与特点

在使用标准函数库中的输入输出函数时，编译系统要求程序提供有关的信息，程序第1行“`#include <stdio.h>`”的作用就是用来提供这些信息的，在此读者不必深究，这在后面章节中有详细的介绍。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，所以在本例中删去第一行的包含命令`#include<stdio.h>`也是正确的。

1.2 C语言的发展与特点

【例1.2】 求两数之和。

```
#include <stdio.h>
main( )          /*求两数之和*/
{
    int a,b,sum;   /*这是定义变量*/
    a=88;
    b=99;
    sum=a+b;
    printf ("sum is %d\n", sum);
}
```

1.2 C语言的发展与特点

本程序的作用是求两个整数a和b之和sum。/* */表示注释部分，为了便于理解，这里采用汉字来注释，当然也可以用英语或汉语拼音来注释。注释可以加在程序的任何位置，它的只是方便对程序语句的理解，在程序编译和运行时不起作用。第4行是变量定义部分，定义a和b为整型（int）变量。第5、6行是两个赋值语句，将整数值88和99分别赋值给变量a和b。第7行是将a+b的结果赋值给变量sum，第8行中“%d”是输出函数的“格式说明”，表示要在这个位置输出一个“十进制整数”。printf函数中括弧内最右端的sum是要输出的变量，现在它的值为187（即88+99之值），因此程序运行的结果是：

sum is 187

1.2 C语言的发展与特点

关于C程序的几点说明

(1) C程序是由函数构成的。一个C源程序至少包含一个函数（**main**函数），也可以包含一个**main**函数和若干个其它函数。因此，函数是C程序的基本单位，被调用的函数可以是系统提供的库函数（例如**printf**和**scanf**函数），也可以是由用户根据需要自己编制设计的函数。

(2) 一个函数由函数的首部和函数体两个部分组成。函数名后面必须跟一对圆括号，但括号中的函数参数可以没有，如**main()**。函数体，即函数首部下面的花括号内的内容。如果一个函数内有多个花括号，则最外层的一对花括号为函数体的范围。

(3) 一个C程序总是从**main**函数开始执行的，而不论**main**函数在整个程序中的位置如何。在程序中**main**函数可以放在程序最前头；也可以放在程序的最后；或在一些函数之前，在另一些函数之后。

1.2 C语言的发展与特点

(4) C程序书写格式自由，一行内可以写多个语句，一个语句也可以分写在多行上。

(5) 每条语句和数据定义的最后必须有一个分号。分号是C语句的必要组成部分。例如：`sum=a+b;` 语句后的分号不可缺省，即使是程序中的最后一条语句也应包含分号。

(6) C语言本身没有输入输出语句，输入和输出操作都是由库函数`scanf`和`printf`等函数来完成的。C对输入输出实行“函数化”。

(7) 可以用`/* */`对C程序中的任何部分作注释。有使用价值的源程序都应当加上必要的注释，以增加程序的可读性。

1.3 C语言的字符集与词汇

1.3.1 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，空格，标点和特殊字符组成。在字符常量，字符串常量和注释中还可以使用汉字或其它可表示的图形符号。

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。在其它地方出现时，只起间隔作用，编译程序对它们忽略不计。因此在程序中使用空白符与否，对程序的编译不发生影响，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

1.3 C语言的字符集与词汇

1.3.2 C语言词汇

在C语言中使用的词汇分为六类：标识符，关键字，运算符，分隔符，常量，注释符等。

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。

1.3 C语言的字符集与词汇

以下标识符是合法的:

a, x, x3, BOOK_1, sum5

以下标识符是非法:

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)

1.3 C语言的字符集与词汇

在使用标识符时还必须注意以下几点：

(1)标准C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中规定标识符前八位有效，当两个标识符前八位相同时，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，以便于阅读理解，作到“顾名思义”。

1.3 C语言的字符集与词汇

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、说明变量、函数或其它数据结构的类型。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1.3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`。

1.3 C语言的字符集与词汇

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。

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，标识符之间必须要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。

1.3 C语言的字符集与词汇

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。在后面章节中将专门给予介绍。

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。

1.4 在VC++6.0环境编译运行C语言程序

怎样在VC++6.0环境下编译运行C语言程序

请参见课本相关内容