

C YUYAN
CHENGXU SHEJI
JIAOCHENG

第二版

21 世纪高等院校规划教材 · 计算机类

C语言程序设计教程

主 编 吴国凤 副主编 宣善立



中国科学技术大学出版社

21 世纪高等院校规划教材 · 计算机类

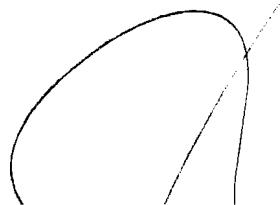
C 语言程序设计教程

第 二 版

主 编：吴国凤

副主编：宣善立

主 审：王 浩



中国科学技术大学出版社

2006 · 合肥

内 容 简 介

本书是面向 21 世纪高等院校规划教材,也是高等院校非计算机专业第一门程序设计课程教材。全书共分 10 章,主要内容包括:C 语言概述、数据类型与运算规则、程序控制结构、数组、函数、指针、结构体与联合、位运算、文件、面向对象及 C++ 基础知识等。

全书内容丰富,系统性强,深入浅出。在结构上突出了以程序设计为中心,以语言知识为工具的思想,对 C 语言的语法规则进行了整理和提炼;在内容上注重知识的完整性;在写法上追求循序渐进,通俗易懂。本书各章均有习题,另配《C 语言程序设计教程上机指导》书,特别适合工科高等院校计算机专业及非计算机专业程序设计语言教科书。

图书在版编目(CIP)数据

C 语言程序设计教程(第二版)/吴国风主编. —合肥:中国科学技术大学出版社,2003.2
ISBN 7-312-01164-0

I.C… II.吴… III.C 语言-程序设计-高等学校-教材 IV.TP312

中国版本图书馆 CIP 数据核字(2003)第 005441 号

中国科学技术大学出版社出版发行

(安徽省合肥市金寨路 96 号,邮编:230026)

合肥学苑印务有限公司印刷

全国新华书店经销

开本:787×1092/16 印张:16.25 字数:410 千

2006 年 2 月第 2 版 2006 年 2 月第 4 次印刷

印数:16101—23100 册

ISBN 7-312-01164-0/TP·311 定价:22.00 元



前 言

C 语言是一种在世界范围内被普遍采用的优秀的程序设计语言。C 语言具有许多优点，主要表现在 C 语言可以用于开发系统软件，它既具有高级语言的可理解性又能直接对硬件操作；C 语言功能完善，可以适用于各种需要；C 语言自身的运算丰富、表达简洁、便于移植，其生成的代码效率高。

作为培养高素质人才的高等院校均已将计算机基础教育放到了很重要的位置。计算机基础教育不仅包括计算机的应用基础，更重要的还包括程序设计基础。对于理工科院校，在程序设计语言的学习中 C 语言是较好的选择。通过程序设计的学习，可以使学习者掌握程序设计的基本概念和一般方法；掌握算法的概念和要求；并建立模块化设计的概念。本教材正是基于这些要求，并综合各位老师的长期教学经验而编写的。

教材全面介绍了 C 语言的语法结构，主要包括基本语言元素；C 语言的基本数据类型、构造类型（数组、结构体与联合）和指针类型等数据表示方法；C 语言的控制结构；模块设计基础的函数以及输入输出文件；同时也简要介绍了面向对象程序设计语言 C++ 的基本概念。

教材组织精练，例题精练，容易理解，并配备了各种类型的练习，便于学习掌握。教材在介绍 C 语言的语法结构的同时，也强调了计算机算法以及结构化设计方法的概念和作用。教材适用于工科高等学校非计算机专业的 C 语言程序设计课程。

本书第 1 章、第 2 章由谢文佩编写，第 3 章、第 4 章由王金玲编写，第 5 章、第 6 章由吴国凤编写，第 7 章、第 9 章由宣善立编写，第 8 章、第 10 章由方乃义编写。全书由吴国凤、宣善立统编定稿，由王浩主审。由于编者水平有限，书中难免有疏忽、错误之处，恳请读者批评指正。

本书在编写过程中，得到兄弟高校计算机基础教育教师的关心和帮助，教研室的同仁们提出了许多宝贵意见；专家顾问们给予了悉心指导；在出版过程中，得到了中国科学技术大学出版社和合肥工业大学教材科的极大帮助，在此一并表示衷心的感谢。

编 者
2003 年 1 月



目 录

前 言	(I)
第 1 章 C 语言概述	(1)
1.1 C 语言的发展简史与特色	(1)
1.2 简单的 C 程序介绍	(3)
1.3 计算机算法与程序设计语言	(5)
习 题	(10)
第 2 章 基本数据类型及其运算	(12)
2.1 标识符、关键字	(12)
2.2 数据类型	(13)
2.3 常量与变量	(14)
2.4 整型数据	(16)
2.5 实型数据	(19)
2.6 字符型数据	(21)
2.7 运算符和表达式	(23)
习 题	(31)
第 3 章 程序控制结构	(33)
3.1 顺序结构	(33)
3.2 分支结构	(42)
3.3 循环结构	(49)
3.4 应用程序举例	(56)
习 题	(62)
第 4 章 数 组	(73)
4.1 一维数组	(73)
4.2 二维数组	(78)
4.3 字符数组	(81)
4.4 应用程序举例	(88)
习 题	(91)
第 5 章 函 数	(97)
5.1 概 述	(97)



5.2	函数间的参数传递	(107)
5.3	数组作为函数参数	(111)
5.4	变量的作用域和存储类型	(115)
5.5	函数的嵌套调用和递归调用	(124)
5.6	编译预处理	(128)
5.7	应用程序举例	(133)
习 题		(138)
第 6 章 指 针		(144)
6.1	指针的基本概念	(144)
6.2	指针与数组	(149)
6.3	指针与函数	(155)
6.4	指针数组与数组指针	(159)
6.5	多级指针	(162)
6.6	带参数的主函数	(164)
6.7	应用程序举例	(165)
习 题		(168)
第 7 章 结构体与联合		(174)
7.1	结构体	(174)
7.2	联合	(191)
7.3	枚举	(194)
7.4	用户自定义类型	(196)
习 题		(197)
第 8 章 位 运 算		(202)
8.1	位运算的概念	(202)
8.2	计算机内的数据表示方法	(202)
8.3	位运算	(204)
8.4	位域(位段)	(208)
习 题		(210)
第 9 章 文 件		(213)
9.1	文件概念	(213)
9.2	文件指针	(214)
9.3	文件的打开与关闭	(215)
9.4	文件的读写	(217)
9.5	文件的定位操作	(224)
9.6	文件的错误检测	(226)
习 题		(227)



第 10 章 面向对象及 C++ 基础知识	(229)
10.1 C++ 简介	(229)
10.2 C++ 的程序结构	(231)
10.3 C++ 的类和对象	(232)
10.4 C 转入 C++ 的一些特性	(236)
习 题	(238)
附录 A C 语言运算符的优先级与结合性	(240)
附录 B 常用字符与 ASCII 代码对照表	(241)
附录 C Turbo C2.0 常用库函数	(242)
附录 D 常见错误信息表	(250)
主要参考文献	(254)



第 1 章 C 语言概述

1.1 C 语言的发展简史与特色

1.1.1 C 语言的发展简史

C 语言是目前世界上广泛使用的计算机高级程序设计语言。适合于作为系统描述语言，既可以编写系统软件（如操作系统），也可以编写各种应用软件，所以在数百种计算机语言中，C 语言仍然是目前最流行、最受欢迎的语言之一。

最初，C 语言是为编写 UNIX 操作系统而设计的。在 C 语言产生之前，操作系统主要是用汇编语言编写的，但是汇编语言依赖计算机硬件，程序的可读性和移植性都比较差。人们一直在寻找一种很接近计算机硬件的高级语言，C 语言就是在这种情况下应运而生的。

C 语言是在 B 语言的基础上发展起来的，它的根源可以追溯到 ALGOL 60。1960 年出现的 ALGOL 60 是一种面向问题的高级语言，它离硬件比较远，不宜用来编写系统程序。1963 年英国剑桥大学在 ALGOL 语言基础上增添了硬件处理能力后，推出了 CPL（Combined Programming Language）语言。CPL 语言在 ALGOL 60 的基础上接近硬件一些，但规模比较大，难以实现。1967 年英国剑桥大学的 Martin Richards 对 CPL 语言作了简化，推出了 BCPL（Basic Combined Programming Language）语言。1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，作了进一步简化，设计出了很简单的 B 语言（取 BCPL 的第一个字母），并用 B 语言编写了第一个 UNIX 操作系统，在 PDP-7 上实现。但 B 语言过于简单，功能有限。1972 年至 1973 年间，贝尔实验室的 D·M·Ritchie 在 B 语言的基础上设计出了 C 语言（取 BCPL 的第二个字母）。C 语言既保持了 BCPL 和 B 语言的优点（精练，接近硬件），又克服了它的缺点（过于简单，数据无类型等）。

最初的 C 语言是为描述和实现 UNIX 操作系统提供一种工作语言而设计的。后来，C 语言多次作了改进，直到 1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们的普遍注意。1983 年，美国国家标准化协会（ANSI: American National Standard Institute），根据 C 语言问世以来各种版本对 C 的发展和扩充，制定了新的标准，称为 ANSI C。1987 年又公布了新标准——87ANSI C。目前流行的各种 C 版本都是以这个标准为基础。

现在使用的各种 C 语言编译系统虽然基本部分是相同的，但也有一些不同。在微型机上使用的有 Microsoft C，Turbo C，Quick C 等，它们的不同版本又略有差异。因此，在使用一



个系统之前，要了解所用的计算机系统配置的 C 编译系统的特点和规定。

1.1.2 C 语言的特色

C 语言能得到快速发展，受到用户的广泛欢迎，是因为它独具特色。

1. C 语言与其他语言的比较

作为高级语言的 C 语言常被称为中级计算机语言。这并不意味着 C 语言功能差，难以使用。C 语言被称为中级计算机语言，只是意味着它把其他高级语言的基本结构与低级语言的实用性结合了起来。

(1) C 与汇编语言比较

汇编语言是非结构化语言，含有大量的跳转、子程序调用以及变址，这种结构的缺陷使得汇编语言程序难以理解，难以维护，也不能移植。

C 语言允许对位、字节和地址（指针）进行操作，这三者是计算机基本的工作单元，在编制系统程序时要经常用到，所以它适用于编写系统程序。C 语言的结构化、模块化克服了汇编语言难读难维护的缺点。另外它又具有汇编语言的功能，目标代码长度相当，效率几乎与汇编语言相近，且具有很好的可移植性。

(2) C 与其他高级语言比较

C 有丰富的运算符，达 34 种，其中有很多运算符对应于常用的机器指令，比如 ++、-- 等运算符可直接编译成机器代码，使用起来简单精炼。

C 有多样化的表达式类型，因此可以将 C 语言说成是表达式语言。C 语言中的表达式几乎无处不在，而在其他高级语言中，表达式则要受到很大的限制。

C 的数据类型丰富，具有现代语言的各种数据结构。C 的数据类型有：整型、实型、字符型、数组、指针、结构体、联合等。它们能用来实现各种复杂的数据结构：链表、树、队列、栈等。其中指针类型使函数参数传递时简单、迅速，节省内存。

C 的输入输出使用的是数据流，而一些高级语言使用的是记录。

C 程序生成的机器代码质量高，内存占用少，运行速度快，程序执行效率高。这是衡量一种语言优劣的重要指标之一。如 FORTRAN 等语言编程生成的代码长，占用内存多，不能用于实时操作。

2. C 是结构化语言

C 语言是以函数为模块来编写源程序的，所以 C 程序是模块化的。

C 语言具有结构化的控制语句，如 if-else 语句，while 语句，do-while 语句，for 语句等。因此是结构化的理想语言，符合现代编程风格的要求。

3. C 是编程者的语言

C 语言简洁、紧凑，使用方便灵活；一共只有 32 个关键字，9 种控制语句，它们构成了 C 语言的全部指令；程序书写形式自由，压缩了一切不必要的成分。

C 语言很少限制、很少缺陷、采用模块结构、彼此独立的函数和一些十分紧凑的关键字，使得 C 语言能够达到汇编语言的高效率和广泛的应用范围，所以在许多情况下它是编程者首选的计算机语言。



4. C语言的“缺点”

(1) 语法限制不严格

C 程序在运行时不做诸如数组边界检查和变量类型兼容性检查，而是由编程者自己保证程序的正确性。故初学者在编程过程中应给予相当的重视，否则很容易发生错误。

(2) 程序设计自由度大

C 语言虽有五种固有的数据类型，但不像 Pascal、Ada 语言有很强的类型区分。实际上 C 语言允许几乎所有数据类型的转换。例如在大部分表达式中，字符型和整型数据都可以自由的混合使用；所有类型均可作逻辑型（非零为‘真’，零为‘假’），还可以把某些类型强制转换为指定类型，编程者可以自由发挥。

以上这些“缺点”实际上在某种程度也是 C 语言的特点，它可以使程序员有更大的自主性，能编出灵活、优质的程序。但对于初学者来说这些特点却不易掌握，似乎是“缺点”，只有在熟练掌握 C 语言程序设计之后，才能体会出其灵活的特性。正是 C 语言的这种灵活性使它的适应性更强，应用范围更大，成为目前最流行的语言。

1.2 简单的 C 程序介绍

C 语言是一种结构化的程序语言，通过掌握 C 语言程序设计可以理解结构化程序设计思想。下面先介绍几个简单的 C 程序，然后从中分析 C 程序的特点。

【例 1.1】输出一条信息。

程序如下：

```
#include <stdio.h>
void main(void)
{
    printf("This is the first C program.\n");
}
```

运行结果如下：

This is the first C program.

在程序中 main() 函数被称作“主函数”，在任何一个 C 程序中都必须有一个且只有一个 main() 函数。前面的 void 是对函数的类型说明，() 中的 void 是函数的参数说明，两个 void 都可省略，具体内容在第 5 章介绍。在 main() 函数中有很多 C 语句，用一对大括号括起来。

在本例的程序中，只有一条 C 语句。这条 C 语句是一个输出函数，作用是向屏幕输出信息。它将双引号之间的内容原样输出到屏幕。“\n”是换行符，它的作用是将光标移到下一行的开始处。

【例 1.2】计算两个数的乘积。

程序如下：

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



```
void main(void )
{
    int x,y,result;           /*定义变量*/
    x=14;
    y=2;                      /*给变量赋值*/
    result=x*y;               /*求积*/
    printf("The result is %d\n",result); /*输出两个数的乘积*/
}
```

运行结果如下:

The result is 28

程序中“/* ... */”表示注释,注释部分不参与也不影响程序的运行,注释只是用来帮助人们阅读和理解程序的,注释部分可以加在程序的任何部分。第 4 行是变量定义,说明了 3 个整型变量;第 5 行是赋值语句,给变量 x 和 y 赋值;第 6 行将 $x * y$ 的结果送入变量 result 中;第 7 行是将结果输出到屏幕,“%d”表示在这个位置上将用一个整型数值代替,本程序中是用整数 28 来代替“%d”这个位置。

C 程序中的语句最重要的一个特点就是每条基本语句的后面都要有一个分号。

【例 1.3】输入两个整数,输出两个数中较小的数。

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

```
void main(void)
```

```
{
```

```
    int x,y,z;
```

```
    scanf("%d,%d",&x,&y);
```

```
    z=min(x,y);
```

```
    printf("\nThe Min number is %d\n",z);
```

```
}
```

```
int min(int x,int y)
```

```
{
```

```
    int z;
```

```
    if(x<y)
```

```
        z=x;
```

```
    else
```

```
        z=y;
```

```
    return(z);
```

```
}
```

若程序输入: 3,5

则运行结果为:

The Min number is 3



这个程序包括两个函数，一个是主函数，另一个是 `imin()` 函数，其作用是求出两个整数 `x`、`y` 中较小的一个。在执行时，先由 `scanf()` 函数从键盘读取两个数据，这时由用户从键盘上输入 3, 5, (↵表示回车) 键。此时 `x` 被赋值 3, `y` 被赋值 5。然后执行第 6 行，将 `x` 和 `y` 的值传入 `imin` 函数中。在 `imin` 函数中经过判断后，`z` 中的值就是两个数的较小值。用 `return` 语句将 `z` 的值返回函数调用处。此时程序又回到第 6 行，将 `imin` 函数的返回值赋给变量 `z`。第 7 行，将变量 `z` 的值输出到屏幕上。

从以上三个例子可以总结出 C 语言程序的几个特点：

(1) C 语言的变量在使用之前必须先定义变量的数据类型，未经定义的变量不能使用。定义变量类型的语句应在可执行语句的前面，如上例中 `main()` 函数中的第一个语句就是变量定义语句，它必须放在第一个执行语句 `scanf()` 的前面。

(2) C 程序是由若干个函数构成。一个 C 程序只能有一个 `main()` 函数，也称主函数。用户也可以根据需要设计自己的函数，并且在主函数中调用它，如同【例 1.3】中的 `imin()` 函数。因此，函数是 C 程序的基本组成单位。C 语言中提供了丰富的函数，被称作库函数。标准 C 中提供了 100 多条库函数，Turbo C 和 MS C 中提供了 300 多条库函数。利用这些系统提供的函数可以非常轻松地编写一些功能强大的程序。C 程序的函数式结构，使得 C 程序非常容易实现模块化，便于阅读和维护。

(3) C 程序总是从 `main()` 函数开始执行，最终在 `main()` 函数中结束。`main()` 函数可以放在程序的任意位置。

(4) C 程序的书写格式比较自由，可以在一行上写若干语句，也可以将一条语句写在多行上。

注意：在 C 语言程序中英文字母是区分大、小写的，在定义变量时，同样的字母的大、小写代表不同的变量，而 C 语言的关键字和基本语句都是用小写字母表示的。

(5) C 语言中没有专门的输入、输出语句。标准的输入和输出操作是通过 `scanf()` 和 `printf()` 等库函数来实现的，这充分体现了 C 语言的函数式结构。

(6) C 程序中可以用 `/* ... */` 对任何部分进行注释，一个好的程序都应有必要的注释以提高程序的可读性。

1.3 计算机算法与程序设计语言

1.3.1 算法 (Algorithm) 的概念

广义地讲算法是解决问题的逻辑步骤，是对特定问题求解步骤的一种描述。简单地说，任何解决问题的过程都是由一定的步骤组成的，把解决问题确定的方法和有限的步骤称为算法。只有通过算法能够描述出来的问题，才能够通过计算机求解。对同一个问题，可以有不同的解题方法和步骤，也就有不同的算法。

计算机算法：是用程序解决问题的逻辑步骤，是指令的有限序列。



计算机算法可分为两大类:

- (1) 数值运算算法: 求解数值;
- (2) 非数值运算算法: 事务管理领域。

正确的算法有三个条件:

- (1) 每个逻辑步骤有可以实现的语句来完成;
- (2) 每个步骤间的关系是惟一的;
- (3) 算法要能终止 (防止死循环)。

1.3.2 算法的基本特征

算法是一个有穷规则的集合, 这些规则确定了解决某类问题的一个运算序列。对于该类问题的任何初始输入值, 它都能机械地一步一步地执行计算, 经过有限步骤后终止计算并产生输出结果。归纳起来, 算法具有以下基本特征:

- (1) 有穷性: 一个算法应包含有限的操作步骤, 而不能是无限的。
- (2) 确定性: 算法中每一个步骤应当是确定的, 而不能含糊、模棱两可。
- (3) 有效性: 算法中每一个步骤应当能有效地执行, 并得到确定的结果。
- (4) 输入: 有零个或多个数据的输入。
- (5) 输出: 有一个或多个数据的输出。

1.3.3 算法的表示

原则上说, 算法可以用任何形式的语言和符号来描述, 通常有自然语言、伪代码、流程图、N-S 图、PAD 图、程序语言等。但通常主要采用自然语言、伪代码、流程图、程序语言四种方法。

1. 用自然语言表示

自然语言可以是中文、英文、数学表达式等。用自然语言表示通俗易懂, 缺点是可能文字过长, 不太严格, 表达分支和循环的结构不很方便。

【例 1.4】如求数列 $1+2+\cdots+m$ 的值 N , 当 $N>10000$ 时结束。

算法表示如下:

- ① $n=0$;
- ② $m=0$;
- ③ m 加 1;
- ④ N 加 m ;
- ⑤ 判 N 是否大于 10000, 如果满足关系结束; 不满足关系继续执行③。

【例 1.5】对一个大于或等于 3 的正整数, 判断它是不是一个素数。

算法表示如下:

- ① 输入 n 的值;
- ② $i=2$;



- ③ n 被 i 除, 得余数 r ;
- ④ 如果 $r=0$, 表示 n 能被 i 整除, 则打印 n “不是素数”, 算法结束; 否则执行⑤;
- ⑤ $i+1 \rightarrow i$;
- ⑥ 如果 $i \leq n-1$, 返回③; 否则打印 n “是素数”; 然后算法结束。

或: ⑥ 如果 $i \leq \sqrt{n}$, 返回③; 否则打印 n “是素数”; 然后算法结束。

以上两个例子都是用自然语言编写的算法。

2. 用流程图表示

流程图是用一些框图来表示各种操作。优点是直观形象, 简单易于理解, 便于修改和交流。ANSI 规定了一些常用的符号, 图 1-1 列出了标准的流程图符号。

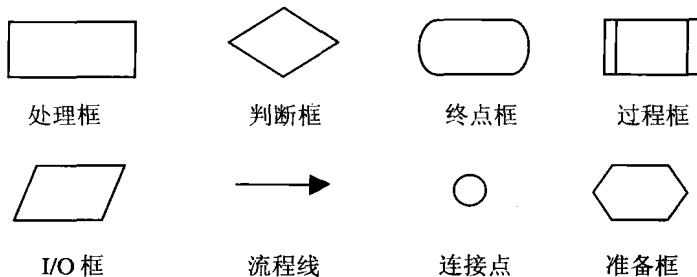


图 1-1 标准流程图符号

【例 1.6】用框图描述如下函数的求解过程, 如图 1-2 所示。

$$y = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

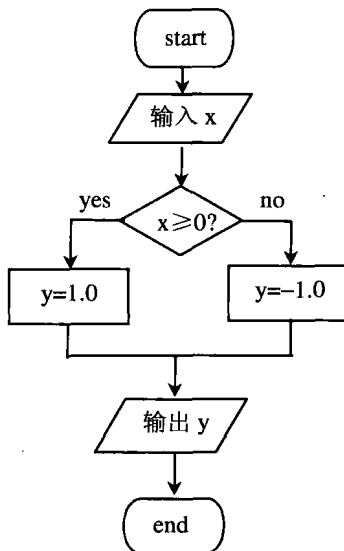


图 1-2 流程图举例



3. 用伪代码表示算法

伪代码是介于自然语言和计算机语言之间，用文字和符号来描述算法的。它不用图形符号，因此，书写方便、格式紧凑、易懂，同时也便于向计算机语言转换。

【例 1.7】用伪代码表示求 $10!$ 的算法。

begin(开始)

置 t 的初值为 1 ($1 \Rightarrow t$)

置 i 的初值为 2 ($2 \Rightarrow i$)

当 (while) $i \leq 10$, 执行 (do)

begin

使 $t = t \times i$

使 $i = i + 1$

end

打印 t 的值 (print t)

end(结束)

4. 用计算机语言表示

计算机是无法识别流程图和伪代码形式，只有用计算机语言编写的程序才能被计算机执行。因此在用流程图和伪代码形式描述出算法后，还要将它转换成计算机语言程序。

【例 1.8】用 C 语言程序来描述求 $10!$ 。

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

```
void main(void)
```

```
{
```

```
    long i,t;
```

```
    t=1L;i=2L;
```

```
    while(i<=10L)
```

```
    {
```

```
        t=t*i;
```

```
        i=i+1;
```

```
    }
```

```
    printf("%ld\n",t);
```

```
}
```

1.3.4 程序设计语言

程序设计语言是用户用来编写程序操作计算机的语言系统。程序设计语言的发展经历了三个阶段：机器语言、汇编语言和高级语言。

1. 机器语言

机器语言是用二进制表示的，计算机能直接识别和执行的语言。机器语言执行速度快，但由于用二进制表示，因而程序可读性差，非专业人员难于掌握。机器语言中的每一条语句



(称为指令)由操作码和操作数组成,反映了操作的对象和性质。

2. 汇编语言

为解决机器语言难于学习和掌握的问题,将二进制指令通过符号化表示就形成了汇编语言。汇编语言采用助记符号表示机器语言中的指令和数据,即用助记符号代替了二进制表示的机器指令,例如用 ADD (Addition) 表示做加法的助记符,每条汇编语言的指令对应一条机器语言的指令。

计算机硬件只能识别机器指令。要执行汇编语言编写的程序,必须先用一个程序将汇编语言的程序翻译成机器语言的程序,用于翻译的程序称为汇编程序。

3. 高级语言

机器语言和汇编语言都是面向机器的。面向机器是指 CPU 不同,机器语言和汇编语言都有可能不一样。从 20 世纪 50 年代中期开始,逐步发展了面向问题(面向过程)和面向对象的程序设计语言,称为高级语言。高级语言表达方式接近被描述的问题,接近于自然语言和数学表达,易于人们接受和掌握。随着计算机技术的发展,形成了为数众多的计算机高级语言,得到广泛应用的有十多种,每种高级语言都有其最适合的应用领域。

如:FORTRAN 语言适用于数值计算;BASIC 语言简单易学,适合于初学者学习;Pascal 语言适用于教学;C 语言适用于系统软件和应用软件的开发;Java 语言既是通用的程序设计语言,又是可以用于 Internet 开发的面向对象的程序设计语言。

由于计算机只能识别机器语言,因此,用高级语言编写的程序(或称源程序)都要通过专用的程序将高级语言翻译成机器语言程序(或称目标程序)后才能被计算机执行。将高级语言翻译成机器语言的方式有两种:一种是将高级语言程序一次性翻译成机器语言,再通过连接程序将系统库连接到程序中形成可执行程序,然后运行,这种称为编译型语言,如图 1-3 所示;另一种是翻译一句执行一句,称为解释型语言。

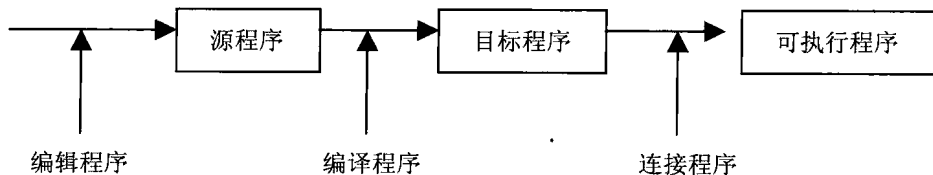


图 1-3 编译型语言的执行过程

C 语言是典型的编译型语言。面向过程的计算机的语言的核心是将输入数据通过算法转换为输出数据。因此,程序的主要描述的是:数据的表示,对数据的处理算法以及数据输入输出。

一、选择题

1. 面向过程的程序设计语言是_____。
A) 机器语言
B) 汇编语言
C) 高级语言
D) 第四代语言
2. C 程序的基本单位是_____。
A) 函数
B) 标识符
C) 表达式
D) 语句
3. 以下_____不是 C 语言的特点。
A) 语言表达能力强
B) 语法定义严格
C) 数据结构系统化
D) 控制流程结构化
4. C 程序是由_____构成的。
A) 主函数与子函数
B) 一个主函数与一个其他函数
C) 主函数与子程序
D) 主函数或主函数与若干个子函数
5. 在 C 语言程序中, main() 函数的位置_____。
A) 必须作为第一个函数
B) 必须作为最后一个函数
C) 可以放在任意位置
D) 必须放在它所调用的函数之后
6. 下面叙述中不正确的是_____。
A) 注释说明只能位于一条语句的后面
B) 注释说明被计算机编译系统忽略
C) 注释说明必须括在“/ * ... */”之间
D) 注释符“/”和“*”之间不能有空格
7. 下面说法中, 正确的是_____。
A) 主函数名是由程序设计人员按照“标识符”的规则选取的
B) 分号和回车符号都可以作为一个语句的结束符号
C) 在程序的构词与构词之间都可以插入一个或多个空格符号
D) 程序的执行总是从源程序清单的第 1 行开始执行

二、填空题

1. 高级语言中的 C 语言, 属于面向_____语言。