

第 1 章 程序设计基础

【教学目标】

- 熟悉程序的基本概念。
- 了解 C 语言的发展及其特点。
- 掌握算法的概念及表示。
- 熟悉 C 语言开发工具。

【重点内容】

- 程序的基本概念。
- 算法及描述。
- C 语言的特点。
- C 语言的开发环境。

1.1 程序及程序设计语言

计算机系统是由硬件和软件两大部分组成的。硬件就是计算机系统中看得见的各种物理部件，是实实在在的器件，是计算机的物质基础；而软件则是指在硬件设备上运行的各种程序及相关资料，是看不见摸不着的，却是计算机的灵魂。没有软件，计算机是一台“裸机”，是什么也做不了的。有了软件，计算机才算有了生命，成为了一台真正的电脑。

一套计算机系统必须是由硬件和软件两者有机结合，才能更好地发挥功效。实际上，在计算机的发展历程中，计算机软件随着硬件技术的快速发展而发展；反过来，软件的不断完善与发展，又促进了计算机硬件的新发展。

1.1.1 什么是计算机程序

计算机程序是用计算机语言描述的为解决某一个问题的步骤指令。每一条指令是计算机执行特定的操作。只要让计算机执行这个程序，计算机才会“自动地”执行各条指令，最终完成某项工作。有人以为计算机是万能的，会自动进行所有的工作，甚至觉得计算机神秘莫测。这是很多初学者的误解。其实，计算机的每一个操作都是根据人们事先指定的指令进行的，离开程序，计算机将一事无成。因此，计算机的本质是程序的机器，程序和指令是计算机系统中最基本的概念。只有懂得程序设计，才能真正了解计算机是怎样工作的，才能更好地使用计算机。

由于计算机不能理解人类的自然语言，所以不能使用自然语言来进行计算机程序编写，只能使用专门设计的程序语言来编写。就像中国人之间用中国话，英国人之间用英语，等等。人们借助计算机能够处理的语言，告诉计算机要处理哪些数据以及按什么步骤来处理，这便是程序设计。

在实际应用中，为解决某一个问题编写的程序不是唯一的，不同的用户编写程序的思路也会不完全一样，因此我们把软件的执行效率、算法、数据结构以及功能性作为评判一个程序的主要因素。

1.1.2 程序设计语言

自 1946 年世界上第一台电子计算机问世以来，计算机科学及其应用的发展十分

迅猛，计算机被广泛地应用于人类生产、生活的各个领域，推动了人类社会的进步。

特别是随着互联网的发展，计算机走进千家万户，为我们的生活带来便捷。在 21 世纪的今天，计算机已经普遍存在于我们生活的每一个角落，从客厅里的智能电视，到随身携带的智能手机，我们的生活正在发生翻天覆地的变化，这些都是计算机的发展所带来的变革。我们目前正处于一个全新的时代——信息化时代。作为一名大学生来说，应该掌握计算机的基础知识和技能，学会利用计算机为自己服务，因此掌握一门编程语言和基本的编程技能是必要的。

计算机语言经历了以下几个发展阶段：

1. 机器语言

机器语言是第一代计算机语言。计算机工作基于二进制，从根本上说，计算机只能识别和接受由 0 和 1 组成的指令，属于低级语言的范畴。

优点：由于机器语言使用的是针对特定型号的计算机语言，故而运算效率是所有语言中最高的。机器语言具有灵活、直接执行和速度快等特点。

缺点：机器语言在使用中比较复杂，编写起来非常烦琐、费时、易出差错，特别是移植性非常差。例如，在一台计算机上执行的程序，想要在另一台计算机上执行，必须重新编写符合本台电脑指令集的程序，造成重复工作。

2. 汇编语言

为了克服机器语言的缺点，人们创造出了汇编语言。汇编语言 (Assembly Language) 是面向机器的程序设计语言。在汇编语言中，用助记符代替操作码，用地址符或标号代替地址码。比如用“ADD”代表加，用“SUB”代表减，等等。这样一来，人们很容易明白程序在干什么，方便纠错和维护。

使用汇编语言编写程序，计算机是不能直接识别的，需要使用一种程序将汇编语言转换为机器指令，这种转换程序我们把它称之为汇编程序。汇编程序是语言处理系统软件，汇编程序把汇编语言翻译成机器语言的过程称为汇编。

虽然汇编语言比机器语言要简单好记一些，但仍然难以普及，只在专业人员中使用。

汇编语言同样十分依赖机器硬件，移植性不好，但效率仍十分高，针对计算机特定硬件而编制的汇编语言程序，能够准确发挥计算机硬件的功能和特长，程序精练而质量高，所以至今仍是一种常用的软件开发工具。由于它贴近计算机，或者说是针对某一台计算机的特性而编写的语言，因此我们也把它称为计算机低级语言。

3. 高级语言

为了克服低级语言的缺点，20 世纪 50 年代创造出了第一个计算机高级语言——FORTRAN 语言。它更接近人们习惯使用的自然语言和数学语言。在这种语言下，其语法和结构更类似普通英文，表示方法要比低级语言更接近于待解决问题的表示方法，

其在一定程度上与具体机器无关，易学、易用、易维护。例如在 FORTRAN 语言程序中，想计算和输出 $3.5 \times 6 \times \sin(\pi/3)$ ，只需写出下面这样一个语句：

```
PRINT*,3.5*6*SIN(3.1415926/3)
```

即可得到计算结果。

这种语言功能很强，且不依赖于具体机器，用它写出的程序对任何型号的计算机都适用（或做出很好的修改），它与具体机器的距离较远，故称为计算机高级语言。当然这种语言机器也不是直接能够识别出来的，也要进行翻译。用一种称为编译程序的软件把程序（源程序）转换为机器指令的程序（目标程序），然后让计算机执行机器指令程序，最后得到结果。高级语言的一个语句往往对应多条机器指令。

自从有了高级语言后，一般的科技人员、管理人员、大中学生以及广大计算机爱好者，都能比较容易地学会用高级语言来进行程序编写，为计算机的普及创造了良好的条件，人们称高级语言的出现为计算机发展史上“惊人的成就”。

数十年来，全世界涌现了 2 500 种以上的高级语言，目前应用最广泛有 100 多种，我们目前大家比较熟悉的有 JAVA（手机应用程序，嵌入式开发），C（系统描述语言），C++（基本程序语言，适用于软件开发），ASP、PHP、JSP（适合动态网站制作），PL/1（大型通用语言）等。

自 20 世纪 80 年代初开始，在软件设计思想上，又产生了一次革命，其成果就是

面向对象的程序设计。在此之前的高级语言，几乎都是面向过程的，程序的执行是流水线式的，在一个模块被执行完成前，人们不能干别的事，也无法动态地改变程序的执行方向，这和人们日常处理事情的方式是不一致的。对人而言是希望发生一件事情就处理一件事情，也就是说，不应面向过程，而应该是面向具体的应用功能，即对象。其方法就是软件的集成化，如同硬件的集成电路一样，生产一些通用的、封装紧密的功能模块，称之为软件集成块，它与具体应用无关，但能相互组合，完成具体的应用功能，同时又能重复使用。对于使用者来说，只需要关心集成块的接口及实现的功能，至于如何实现，使用者完全不用关心，C++、VB、Delphi 就是典型的代表。

高级语言的下一个发展目标面向应用，也就是说，只需要告诉程序你要干什么，程序就能自动生成算法，自动进行处理，这就是非过程化的程序语言。

1.2 C 语言的发展历程及其特点

1.2.1 C 语言的发展历史

C 语言是一种计算机高级语言，在国际上广泛流行。它是 20 世纪 70 年代初在美国贝尔实验室研制出来的，后来又被多次改进。在一段时间里，C 语言还只是在贝尔实验室内部使用，直到 1975 年才引起人们的注意。1977 年出现了可移植的 C 语言，1978 年第七版的 C 语言成了当时的标准，称为标准 C。1983 年，美国国家标准化协

会 (ANSI) 对 C 语言制定了新的标准, 称为 ANSI C。目前流行的 C 语言编译系统大多是以 ANSI C 为基础进行开发的。

C 语言是一种用途广泛、功能强大、使用灵活的过程性 (Procedural) 编程语言, 既可用于编写应用软件, 又能用于编写系统软件。因此, C 语言问世后得到了迅速推广。自 20 世纪 90 年代初以来, C 语言在我国开始推广, 学习和使用 C 语言的人越来越多, C 语言便成为学习和使用人数最多的一种计算机语言, 绝大多数理工院校都开设了 C 语言程序设计课程。熟练掌握 C 语言也就成为计算机开发人员的一项基本功。

由于开发大型软件的需要, 现在 Visual C++ 语言在我国逐步得到推广, 但我们不能说 C 语言过时了。C++ 是面向对象的程序设计, C 语言是面向过程的程序设计, 面向对象的基础是面向过程。因此掌握 C 语言后, 再学习其他编程语言就比较容易了。

1.2.2 C 语言的特点

C 语言之所以能够存在和发展, 并具有较强的生命力, 是因为其具有优越于其他语言的特点。C 语言的主要特点如下:

1. 语言简洁、紧凑, 方便、灵活

C 语言只有 32 个关键字、9 种控制语句, 程序书写形式自由, 主要用小写字母表示, 压缩了一切不必要的成分。采用 C 语言编程可以使程序员专注算法设计, 而不必过多地考虑格式的限制。

2. C 语言具有丰富的运算符

C 语言的运算符包含的范围很广泛，共有 34 种，灵活使用这些运算符可以实现其他高级语言中难以实现的运算。

3. 数据类型非常丰富

C 语言提供的数据类型包括：整型、浮点型、字符型、数组类型、指针类型、结构体类型和共用体类型等，能用来实现各种复杂的数据结构的运算。尤其是指针类型数据，使编程更加灵活、多样。

4. C 语言是完全模块化和结构化的语言

具有结构化的控制语句（如 if-else 语句、while 语句、switch 语句、for 语句）。用函数作为程序的模块单位，便于实现程序的模块化。C 语言是良好的结构化语言，符合现代编程风格要求。

5. C 语言语法限制不是很严格，程序设计自由度大

一般的高级语言语法检查比较严格，能检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度，因此，放宽了语法检查。程序员应当仔细检查程序，保证其正确，而不要过分依赖 C 编译程序去查错。

6. C 语言允许直接访问物理地址

在 C 语言中，可以直接访问到数据存放的物理地址，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 C 语言既具有高级语言的功能，又具有低级语言的许多功能，可以用来编写系统软件。C 语言的这种双重性，使它既是成功的系统描述语言，又是通用的程序设计语言。

7. 与汇编语言相比，可移植性好

C 语言编写的程序可移植性好，基本上不做修改就能用于各种型号的计算机和各种操作系统。

8. 生成目标代码质量高，程序执行效率高

C 语言一般只比汇编程序生成的目标代码效率低 10%~20%。C 语言原来是专门为编写系统软件而设计的，许多大的软件都用 C 语言编写，这是因为 C 语言的可移植性好和硬件控制能力高，表达和运算能力强。许多以前只能用汇编语言处理的问题，现在可以改用 C 语言来处理了。

C 语言的优点很多，但也有一些不足，如运算符优先级太多，记忆起来有一定困难，有些还与常规的认知不同；另外语法限制不太严格，类型检验太弱，不同类型转换比较随意，因此不太安全。

对 C 语言以上的特点，待学完 C 语言以后再来回顾一下，就会有比较深的体会。对于初学者，掌握 C 语言的基本语法规则，多上机练习是学习的要诀。

1.3 简单的 C 程序介绍

为了使用 C 语言编写程序，必须了解 C 语言，并且能够熟练地使用 C 语言。本书将由浅入深地介绍怎样阅读 C 语言程序和使用 C 语言编写程序。下面先介绍几个简单的 C 程序，然后从中分析 C 程序的结构。

【例 1.1】要求在屏幕上输出以下信息：

```
Hello word!
```

编写程序：

```
/*这是第一个程序*/  
  
#include<stdio.h>  
int main ()  
{  
    printf("hello world!\n");  
    return 0;  
}
```

程序运行结果如图 1.1 所示。

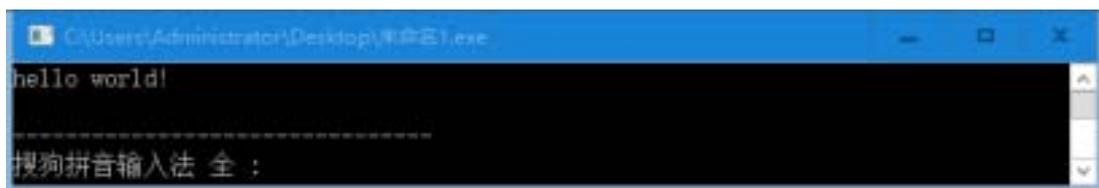


图 1.1 程序运行结果

源程序分析：

(1) 这个 C 程序的第一行是注释语句，`/*...*/`表示注释的部分。注释是对程序的功能进行说明，为便于理解，我们用汉字表示注释，当然也可以用英语或者汉语拼音作注释。注释只是给人看的，对编译和运行不起作用。注释可以加在程序中

的任意位置。

(2) 第 2 行是一个文件包含语句，`#include` 语句是文件包含语句，它指的是一个程序把另一个指定文件的内容包含进来（在后续章节将详细介绍）。`stdio.h` 是一个被包含的文件名字，它是一个“标准输入输出”文件。把文件 `stdio.h` 包含到程序中的作用是可以在程序中实现数据的输入和输出，第 5 行的 `printf()` 函数的头文件就是 `stdio.h`，这样该函数的输出才能被执行。

(3) 第 3 行是主函数 `main()`，每个 C 程序都必须有一个 `main` 函数，它是 C 程序的入口，因为每个可执行的 C 程序总是从 `main()` 函数开始执行的。这是使用 `int main()`，要求 `main` 的返回值为 0，即 `return 0`。

(4) 第 4 行的“{”和第 6 行的“}”及它们所包含起来的部分是 `main` 函数开始执行和结束的标志。

第 5 行的 `printf()` 函数是 C 语言中的输出函数（详见第 2 章）。双引号内的字符串原样输出。“`\n`”是换行符，即在输出“`hello world !`”后回车换行。语句最后有一个分号。

说明：C 语言允许两种注释方式：

(1) 以 `//` 开始的单行注释。如上介绍的注释，这种注释可以单独占一行，也可以出现在一行中其他内容的右侧。此中注释的范围从 `//` 开始，以换行符结束。也就是说这种注释不能跨行。如果注释内容一行内写不下，可以用多个单行注释，如下面两行

是连续的注释行：

```
//如注释内容一行内写不下
```

```
//可以再下一行重新用“//”，然后继续写注释。
```

(2) 以/*开始，以*/结束的对称式注释。这种注释可以包含多行内容。它可以单独占一行（在开头以/*开始，结尾以*/结束），也可以包含多行。编译系统在发现一个/*后，会开始找注释结束符*/，并把两者间的内容作为注释。

但应该注意的是在字符串的//和/*都不作为注释的开始，而是作为字符串的一部分，如：

```
Printf("//how do you do !\n");
```

或

```
Printf("/*how do you do !*/\n");
```

输出的分别是：

```
// How do you do !
```

和

```
/*How do you do !*/
```

注释可以用汉字或者英文字符表示。

【例 1.2】求两个整数之和。

编写程序：

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

```
int main ()
{
    int a,b,c;
    a=789;
    b=456;
    c=a+b;

    printf("两数之和为%d\n",c);

    return 0;
}
```

运行结果如图 1.2 所示。



图 1.2 程序运行结果

源程序分析：

- (1) 本程序中的作用是求两个整数 a 和 b 的和 c。
- (2) 第 4 行是声明部分，定义变量 a,b 和 c，并指定为整型 (int) 变量。
- (3) 第 5、6 行是赋值语句，使 a 和 b 的值分别为 789 和 456。
- (4) 第 7 行是一个表达式语句，用于计算两个数的和，即 c 的值为 a+b。
- (5) 第 8 行中的“%d”是输入输出的“格式字符串”(图 1.3)，用来指定输入输出时的数据类型和格式 (详见第 2 章)。printf 函数中括号内最右侧 c 是要输出的变量。

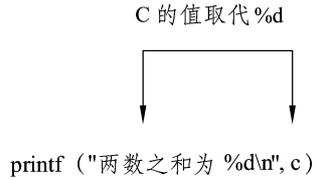


图 1.3 格式控制符

1.4 C 语言程序的基本结构及书写规则

一个完整的 C 语言程序是由一个 main() 函数 (又称主函数) 和若干个其他函数组合而成的 , 或仅由一个 main() 函数构成。下面我们具体通过几个简单的实例来介绍 C 语言程序的基本构成和书写格式 , 使读者对 C 语言程序有一个基本的了解。在此基础上进一步掌握 C 语言程序的语法和书写规则。

1.4.1 C 语言程序的基本结构

【例 1.3】 由一个 main() 函数构成的 C 语言程序。

```
#include<stdio.h>                //头文件

int main ()                        //主函数
{
    printf("hello world!\n");      //输出内容
    return 0;
}
```

程序运行结果如图 1.4 所示。

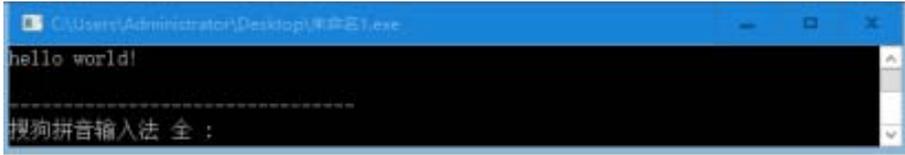


图 1.4 程序运行结果

【例 1.4】由 main()函数和子函数 max()构成的 C 语言程序。

```
#include<stdio.h>

int main( )                //主函数
{
    int max( );            //子函数声明

    int num1,num2;        //变量声明

    printf("请输入第一个数字：");    //输出提示字符

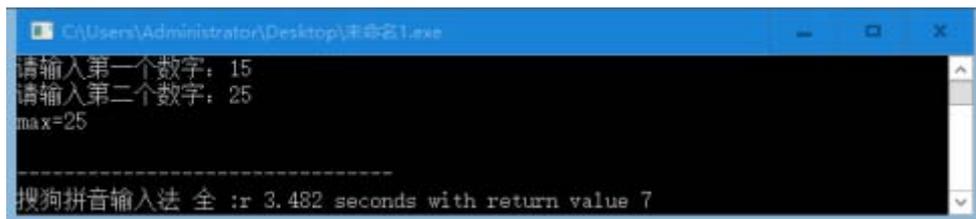
    scanf("%d",&num1);        //输入数值

    printf("请输入第二个数字：");    //输出提示字符

    scanf("%d",&num2);        //输入数值

    printf("max=%d\n",max(num1,num2));    //输出结果
}
int max (int x,int y)
{
    return (x>y?x:y);    //比较两数大小
}
```

程序运行结果如图 1.5 所示。



```
C:\Users\Administrator\Desktop\1.exe
请输入第一个数字: 15
请输入第二个数字: 25
max=25
-----
搜狗拼音输入法 全 :r 3.482 seconds with return value 7
```

图 1.5 程序运行结果

以上两个示例中，例 1.3 的 C 语言程序仅由一个 `main()` 函数构成；例 1.4 所示的 C 语言程序由一个 `main()` 函数和用户自定义的 `max()` 函数构成。由此可见，一个完整的 C 语言程序有以下特点：

- (1) 仅有一个 `main()` 函数构成。
- (2) 由一个且只能有一个 `main()` 函数和若干个其他函数结合而成。其中，自定义函数由用户自己设计。
- (3) 函数是 C 语言程序的基本单位，每一个函数完成相对独立的功能。`main()` 函数相当于主程序，其他函数相当于子程序。
- (4) C 程序中包含预处理命令，预处理命令通常放在程序的最开始位置。如 `#include<stdio.h>`，C 编译系统在对源程序进行编译之前，先由一个“预处理器”对预处理指令进行预处理，对于 `#include<stdio.h>` 指令来说，就是将 `stdio.h` 头文件的内容读取进来。
- (5) `main` 是函数名，函数后面的一对圆括号“()”是用来写函数的参数的。参数可以有，也可以没有，但圆括号不能省略。“{ }”括起来的是函数体部分，包含了实现

函数功能的语句。

(6) 一个 C 语言程序总是从 `main()` 函数开始执行，在 `main()` 函数中结束。主函数执行完毕，程序执行结束。

(7) 程序中对计算机的操作是由函数中的 C 语句完成的，如赋值、输入输出数据的操作都是由相应的 C 语句实现的。C 程序书写格式比较自由，一行内可以写几个语句，一个语句也可以写在多行上，但为了清晰可见，习惯上每一行只写一个语句。

(8) 在每个数据声明和语句的最后必须有一个分号。分号是 C 语句的必要组成部分。如

```
C=A+B;
```

其中的分号是必不可少的。

(9) C 语言本身不提供输入输出语句。输入和输出操作是由库函数 `scanf` 和 `printf` 等函数来完成的。C 对输入输出实行“函数化”。

(10) 程序应当包含注释。一个好的、有使用价值的源程序都应当加上必要的注释，以增强程序的可读性。

1.4.2 源程序书写格式

为了便于阅读和维护，更好地体现程序的层次结构，C 语言的书写应遵循相应的规则。