

程序设计基础篇

第 1 章 C 语言概述

计算机发展到今天，其应用已经深入到许多领域。程序设计所用的计算机语言也从最早面向计算机的低级语言，发展到今天的面向对象的高级语言。

1.1 程序设计语言的发展过程

计算机的一切操作都是由程序控制的，离开程序，计算机将一无所用。程序就是用计算机语言编写的命令序列。计算机语言的种类有很多，根据其发展的过程和面向的对象，可分为三大类：机器语言、汇编语言、高级语言。

1.1.1 机器语言(第一代语言)

机器语言是由二进制代码 0 和 1 构成的指令序列，是面向计算机 CPU 系统的，是计算机可以直接识别并执行的计算机语言。不同的计算机 CPU 系统能够识别的机器语言是不同的。

例如：在某种计算机 CPU 系统中，加法指令用 00100101 表示，减法指令用 10010101 表示。

优点：机器语言能被计算机的 CPU 直接理解和执行，不需要另外的翻译软件，占用空间少，执行速度快。

缺点：机器语言的缺点主要表现在难理解、难编写、难修改、难移植几个方面。

1.1.2 汇编语言(第二代语言)

为了克服机器语言的缺点，人们采用助记符和符号地址来代替机器指令，所形成的计算机语言，称为汇编语言，汇编语言是符号语言。

例如：用 ADD 代表加法，用 SUB 代表减法。

用汇编语言编写的程序，计算机硬件不能直接理解和执行，需要通过另外的翻译软件(汇编程序)将其翻译成机器语言目标程序后，计算机才可以执行。

优点：执行效率高，与机器语言相比，其学习和记忆难度有所下降。

缺点：仍然是面向计算机硬件系统的语言，通用性较差，用户较难掌握，仍然属于计算机的低级语言。

1.1.3 高级语言

高级语言是由表示不同意义的英文单词和数学符号按照一定的逻辑关系及严格的语法规则构成的程序设计语言。

例如：“+”代表加法，“-”代表减法。

高级语言接近于自然语言，便于用户学习和记忆，且通用性较强。

用高级语言编写的程序，计算机硬件也不能直接理解和执行，需要另外的语言处理程序将

其翻译成机器语言程序。高级语言可分为面向过程的高级语言和面向对象的高级语言。

1. 面向过程的高级语言(第三代语言)

面向过程的高级语言在程序中不仅要告诉计算机“做什么”，还要告诉计算机“怎么做”。即在程序中要详细描述用什么动作加工什么数据，也就是把解题过程编写成高级语言程序。

常用的面向过程的高级语言有 BASIC、FORTRAN、Pascal、C、COBOL 等。

2. 面向对象的高级语言(第四代语言)

面向对象的高级语言是非过程化的语言，是面向应用层的。编写程序时，在程序中只需告诉计算机“做什么”，一般无须告诉计算机“怎么做”。

常用的面向对象的高级语言有 Visual Basic、C++、Visual C++、Visual J++、Visual FoxPro、Borland Delphi 和 Power Builder 等。

1.2 C 语言简介

1.2.1 C 语言的发展

C 语言是面向过程的高级语言中的一种，是在 20 世纪 70 年代初问世的。1978 年，美国电话电报公司(AT&T)贝尔实验室正式发布了 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，该书通常简称为 *K&R*，也有人称之为 *K&R* 标准。但是，在 *K&R* 中并未定义一个完整的标准 C 语言；美国国家标准学会(American National Standards Institute)后来在此基础上制定了一个 C 语言标准，并于 20 世纪 80 年代初期发布，该标准通常简称为 ANSI C。

早期的 C 语言主要用于 UNIX 系统。后来，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，C 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了广泛使用，成为当代最优秀的程序设计语言之一。

目前最流行的 C 语言有以下几种版本。

- (1) Visual C++，简称 VC++。
- (2) Microsoft C，简称 MS C。
- (3) Borland Turbo C，简称 Turbo C。
- (4) AT&T C。

这些 C 语言版本都在 ANSI C 标准的基础上各自做了一些扩充，使之更加方便、完美。20 世纪 80 年代，贝尔实验室又为 C 语言增加了面向对象的特性，即 C++。

1.2.2 C 语言的特点

1. 简洁紧凑

C 语言一共有 32 个关键字、9 种控制语句，一般用小写字母表示。由于压缩了一切不必要的內容，所以完成同样功能的程序更短小精悍。

2. 数据类型齐全

C 的数据类型有：整型、实型、字符型、数组型、指针型、结构体型、共用体型等，几乎包含了其他高级语言中的所有数据类型，能方便地实现各种不同数据类型的运算。另外，又引入了能实现如链表、树、栈等复杂数据结构的指针概念，使程序更加灵活和多样化、效率更高。

3. 运算符丰富

C 语言共有 34 种运算符。C 语言把括号、赋值等都作为运算符处理，从而使 C 语言的运算类型极其丰富，灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

4. C 语言是结构化语言

结构化语言的显著特点是代码及数据的分隔化，即程序的各个部分除了必要的信息交流外，彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护及调试。C 语言是以函数形式提供给用户的，用户可以方便地调用这些函数。另外，C 语言提供了多种结构化的循环、条件控制语句，从而使程序很容易地实现完全结构化。

5. C 语言语法限制不太严格、程序设计自由度高

大多数高级语言对程序语法检查比较严格，能够检查出几乎所有的语法错误。而 C 语言放宽了对程序语法的检查，程序员用 C 语言写程序会感到限制少、灵活性大，允许程序编写者有较大的自由度。

6. C 语言可以直接对硬件进行操作

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，能够像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元。C 语言功能强大，既可以用于编写应用软件，也可用于编写系统软件，因此被称为“高级语言中的低级语言”或“中级语言”。

7. C 语言程序可移植性好、生成代码质量高、程序执行效率高

C 语言的一个突出的优点是适用于多种操作系统，也适用于多种机型。用 C 语言编写的程序几乎不用修改就能用于其他各种型号的计算机和各种操作系统，可移植性非常好。另外，C 语言编写的程序生成的代码质量高，只比汇编程序生成的目标代码效率稍低一些，从而使程序执行效率较高。

当然，C 语言也有缺点，如灵活性给编程人员带来自由的同时，可能也埋下了一定的风险；指针使得程序的执行过程难以跟踪；简洁使得程序难以阅读等，但与其众多的优点相比，仍然不失为人们首选的编程语言之一。

1.2.3 C 语言和 C++ 语言

20 世纪 80 年代，贝尔实验室推出了 C++，它是在 C 语言的基础上发展而来的，进一步扩充和完善了 C 语言的各项功能，成为一种面向对象的程序设计语言。C++ 目前流行的版本有 Borland C++、Symantec C++、Microsoft Visual C++。

C++ 提出了一些全新的概念，所支持的这些新的面向对象的概念容易将问题空间直接地映射到程序空间，为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。就 C++ 来说，语言的复杂性增加了很多，掌握起来有一定难度。

C 是 C++ 的基础，C++ 语言在很多方面和 C 语言是兼容的。因此，掌握了 C 语言，再进一步学习 C++ 就能以一种熟悉的语法来学习面向对象的语言，对学好 C++ 会有较大的帮助。

1.3 C 语言程序的结构

1.3.1 简单的 C 语言程序

阅读以下程序并从中了解一个 C 语言源程序的基本组成和书写格式。

【例 1-1】 输出一个字符串。

```
/* exp1-1 */
#include "stdio.h"
int main( )
{
    printf("C 语言是优秀的程序设计语言\n");
    return 0;
}
```

程序运行结果：

```
C语言是优秀的程序设计语言
Press any key to continue_
```

程序分析：

(1) `/* exp1-1 */`

C 语言程序的注释语句，内容不影响程序的执行，只是为了帮助阅读和理解程序。注释内容可用汉字、英文等各种符号。

(2) `#include "stdio.h"`

文件包含命令，当在源程序中需调用库函数时，要将该函数所对应的头文件包含到该程序中，`stdio.h` 文件为输入和输出头文件。除此之外，常用的库函数头文件还有数学函数的头文件 `math.h`、字符串函数的头文件 `string.h` 等。

(3) `int main()`

`main`：C 语言程序的主函数名；`int`：定义函数类型为整型。

(4) `{`

C 语言程序的函数体开始标志。

(5) `printf("C 语言是优秀的程序设计语言\n");`

`printf` 为格式输出函数，其功能是在屏幕上输出双引号中的一行信息。`\n` 实现换行。

(6) `return 0;`

`return` 为返回函数值语句，该语句的功能是验证主函数是否正常执行。若主函数正常执行，返回值为 0。

(7) `}`

C 语言程序的函数体结束标志。

【例 1-2】 求三个数的平均数。

```
/* exp1-2 */
#include "stdio.h"
int main( )
{
    int x, y, z; /* 定义 x, y, z 为整型变量 */
    float average; /* 定义 average 为实型变量 */
    x = 5; /* 给变量 x 赋初值 */
    y = 12; /* 给变量 y 赋初值 */
    z = 35; /* 给变量 z 赋初值 */
    average = (x + y + z) / 3.0; /* 计算平均数 */
    printf("average = %f\n", average); /* 输出平均数 */
    return 0;
}
```

程序运行结果:

```
average = 17.333333
Press any key to continue
```

程序分析:

(1)/* exp1-2 */

C 语言程序的注释语句。

(2)#include "stdio.h"

文件包含命令。

(3)int main()

main: C 语言程序的主函数名; int: 定义函数类型为整型。

(4){

C 语言程序的函数体开始标志。

(5)int x,y,z;

定义 3 个变量, 而且数据类型是整型。

(6)float average;

定义 1 个变量表示平均数, 而且数据类型是实型。

(7)x = 5;

y = 12;

z = 35;

此 3 行为赋值语句, 给 3 个变量分别赋初始值 5、12、35。

(8)average = (x + y + z) / 3.0;

计算 x、y、z 的平均值。

(9)printf("average = %f \n", average);

printf 为格式输出函数, 在屏幕上输出平均数。其中%f 是格式控制符, 控制变量 average 的值以实型数据格式输出(将在本书第 3 章学习)。

(10)return 0;

return 为返回函数值语句, 该语句的功能是验证主函数是否正常执行。若主函数正常执行, 返回值为 0。

(11) }

C 语言程序的函数体结束标志。

【例 1-3】 求两个数中的大数。

```
/* exp1-3 */
#include "stdio.h"
int main( )
{
    int max(int x, int y);          /* 声明要调用的 max 子函数 */
    int a, b, c;                  /* 定义 3 个整型变量 a,b,c */
    scanf("%d%d", &a, &b);       /* 由键盘输入两个数分别给 a,b */
    c = max(a, b);                /* 调用 max 函数求两个数中的大数 */
    printf("最大数是:%d\n", c);   /* 输出两个数中的大数 */
    return 0;
}
```

```

int max(int x, int y)                /* 定义 max 子函数 */
{
    int z;                          /* max 函数中的声明部分 */
    if(x > y)                        /* 比较两个数的大小 */
        z = x;                       /* x,y 中的大数 x 赋给变量 z */
    else
        z = y;                       /* x,y 中的大数 y 赋给变量 z */
    return z;                        /* 返回两个数中的大数 */
}

```

程序运行结果:

```

-8 12
最大数是:12

```

程序分析: 本程序中的一些语句行的功能与前两个程序完全一样, 在这里不再分析。下面只对程序中的部分语句行进行分析。

(1) `int max(int x, int y);`

对被调函数 `max` 的声明, 告知编译系统主函数的执行中将会调用 `max` 函数。该函数的类型为整型(返回值为整型), 并有两个整型参数。

(2) `scanf("%d%d", &a, &b);`

`scanf()` 为格式输入函数, 从键盘给变量 `a` 和 `b` 输入数值。

(3) `c = max(a, b);`

调用函数 `max` 的语句, 调用时会将实参 `a` 和 `b` 的值传递给 `max` 函数中的形式参数 `x`, `y`, 并将 `max` 函数中的返回值赋给变量 `c`。

(4) `return z;`

返回函数值语句, 将 `max` 函数中求得的最大数返回到主函数中。

本程序将要实现的功能分给两个函数来完成, 主函数(`main` 函数)负责数据的输入和输出; 子函数(`max` 函数)负责在两个数中找出最大数。在主函数中给 `a` 和 `b` 两个变量输入数据, 并通过函数调用语句将变量 `a` 和 `b` 的值传递给子函数中的变量 `x` 和 `y`; 在子函数中, 对变量 `x` 和 `y` 进行判断, 将较大的那个值放入变量 `z` 中, 并将变量 `z` 的值返回给主函数, 由主函数负责输出, 即两个函数共同合作完成任务。

1.3.2 C 语言程序的结构

以上几个程序, 虽然还不能完全包含 C 语言程序的全部, 但从中已可以了解一个 C 语言程序的基本结构和书写格式。

1. C 语言程序由一个或若干个函数组成

(1) C 语言程序由一个或若干个函数组成, 其中有且仅有一个函数名为 `main` 的主函数。

(2) 无论主函数写在什么位置, C 语言程序总是从主函数开始执行, 结束于主函数。

(3) 被调用的函数可以是系统提供的库函数, 如 `printf()`、`scanf()` 等, 也可以是用户自定义的子函数, 如 `max()`。如果调用系统提供的库函数, 在调用之前必须将相应的头文件包含到本程序中, 如果调用用户定义的子函数, 在调用之前必须声明。

例如，上面程序中的语句`#include "stdio.h"`和`int max(int x, int y);`，就起这样的作用，其详细内容将在后续的章节中介绍。

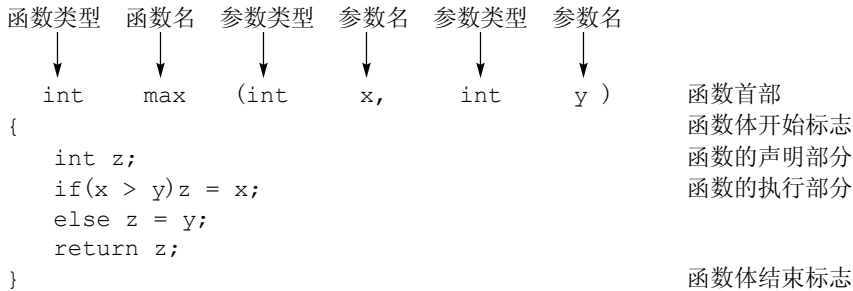
(4) C 语言程序的函数相当于其他语言中的子程序，用函数来实现某特定功能，编写 C 语言程序实际上就是编写一个个函数。

2. C 语言程序中的每一个函数由两部分组成

在 C 语言程序中，每一个函数又由函数首部和函数体两部分组成。

(1) 函数首部，即函数的第一行，包括函数类型、函数名、函数参数类型、函数参数名。

(2) 函数体，即函数首部下面花括号内的部分，由说明和执行两部分组成。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。下面以子函数 `max` 为例来看一下函数的构成。



3. 函数体由若干 C 语句组成

(1) 函数体由若干 C 语句组成，C 语句有多种类型。例如：`/* …… */`为注释语句；`int z;`为声明语句；`z = x;`为赋值语句；`return z;`为返回语句等。

(2) C 语句必须以分号(;)作为语句结束符。

(3) C 语句书写比较自由。例如，可在一行书写多条语句，也可将一条语句写在多行，但习惯一行只写一条语句。

4. C 语句由一些基本字符组成

C 语言程序的语句中含有各种符号、名称、数值等。例如，`int` 是英语单词 `integer`(整型)的缩写，表示整型；`average=(x+y+z)/3.0;`类似于数学式子；`x>y` 类似于不等式；`3.0` 代表一个数值；`return z` 代表的功能也与英文原意相同等。

5. C 语言程序区分大小写字母

由于在 C 语言程序中区分大小写字母，因此命名时应特别注意。

6. C 语言对输入/输出实行“函数化”

C 语言系统本身没有设置输入/输出语句，输入/输出的操作是通过调用函数库中的 `scanf()`、`printf()` 等函数来完成的。

1.4 C 语言程序的运行步骤和集成开发环境

1.4.1 C 语言程序的运行步骤

编好一个 C 语言程序后，如何上机运行呢？在所选用的系统上，一般要经过如下几个步骤。

1. 输入和编辑源程序

在启动所选用的 C 语言集成环境之后，在源代码编辑窗口，将程序输入到计算机中，并对源程序进行编辑修改，借助于编程环境建立 C 语言程序的源代码，并形成代码文件，其扩展名为“.c”。例如，编辑后得到一个源程序文件“f.c”。

2. 编译程序

计算机系统无法识别用 C 语言编写的高级语言的源程序，必须将 C 语言程序的源代码文件转换为用机器语言表示的目标代码文件，扩展名为“.obj”。例如，编译后得到一个目标代码文件“f.obj”。

3. 连接程序

C 语言程序中引用了一些库函数，如 `printf()` 和 `scanf()` 等，系统还必须从系统的库中抽取引用的库函数的代码，将其加入到本程序代码中，使得各程序模块结合为一个有机的整体，最终形成计算机可以理解的、可直接运行的可执行程序，其扩展名为“.exe”。例如，连接后得到一个可执行文件“f.exe”。

4. 运行程序

得到可执行文件后，将该文件调入内存并使之执行，即可按照程序的要求得到程序的结果。

程序编写过程中，若在编辑和编译阶段发现错误，需要对程序代码的语法进行修改，若在运行程序时发现结果错误，需要对程序代码的逻辑进行修改。上述过程可能需要反复很多次才能使程序得到正确的运行结果，这就是程序的调试，调试就是发现并修正错误。通过不断地查找错误，修正错误，最终得到正确的运行结果。这个过程中，程序员的编程能力也在同步增长，编程水平会不断提高。

以上 C 语言程序的运行步骤可形象地表示为图 1-1。

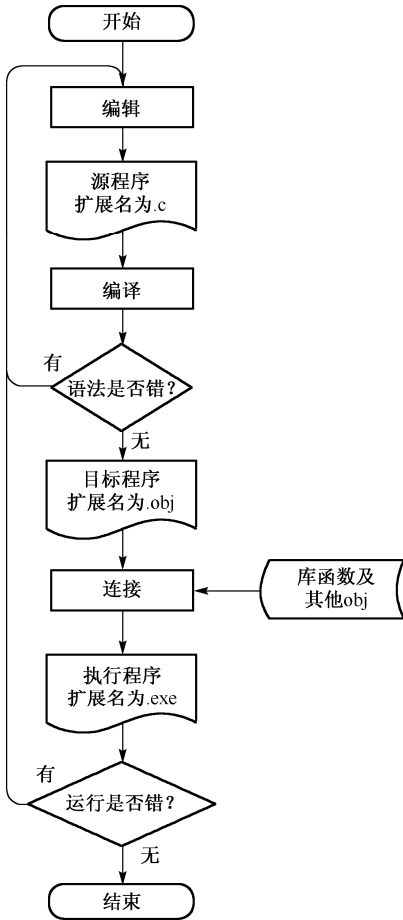


图 1-1 C 语言程序运行步骤

1.4.2 C 语言的集成开发环境

编译、连接、运行源程序需要有相应的编译系统。目前使用的大部分编译系统都是集成开发环境，它把程序的编译、连接、运行操作全部集成在一个界面上来进行，功能齐全、使用方便直观。常用的编译系统有 Turbo C 2.0、Turbo C++ 3.0、Visual C++ 6.0 等。

1. Turbo C 2.0

Turbo C 2.0 以往用得比较多，它是在 DOS 操作系统下使用的，由于只能用键盘操作，而不能使用鼠标操作，再加上不能使用复制、剪切、粘贴等功能，因此感觉很不方便，现在已不常使用了。

2. Turbo C++ 3.0

近年来，有比较多的人使用 Turbo C++ 3.0，尽管 Turbo C++ 3.0 也是在 DOS 环境下运行的，

但用户可以在 Windows 操作系统下直接运行 Turbo C++ 3.0 编译系统，其使用较之 Turbo C 2.0 更为方便。

3. Visual C++ 6.0

Visual C++ 6.0 是 Windows 下的集成环境，支持复制、剪切、粘贴等功能，能够使用鼠标操作。这些年随着 C++ 语言程序的普及，Visual C++ 6.0 作为一种功能强大的程序编译器也被相当多的程序员所使用。由于 Visual C++ 6.0 对 C 语言也是兼容的，所以也可以选用它作为学习 C 语言的集成环境。这也有利于同学们以后进一步学习 C++ 语言。Visual C++ 6.0 平台下运行 C 语言程序的过程将在本书配套实验指导书中详细介绍。

小 结

本章主要介绍了程序设计语言的发展、分类，C 语言的基本特点、构成规则，C 语言程序的上机运行步骤。通过本章的学习，读者可对计算机程序设计语言，特别是 C 语言有一个总体认识。

习 题 1

一、选择题

1. 以下叙述正确的是_____。
 - A) 在 C 语言程序中，主函数必须位于程序的最前面
 - B) 在 C 语言程序中，一行只能写一条语句
 - C) C 语言程序的基本结构是程序行
 - D) C 语句是完成某种程序功能的最小单位
2. 一个 C 程序的执行是从_____。
 - A) 本程序的主函数开始，到本程序的主函数结束
 - B) 本程序的第一个函数开始，到本程序的最后一个函数结束
 - C) 本程序的主函数开始，到本程序的最后一个函数结束
 - D) 本程序的第一个函数开始，到本程序的主函数结束
3. 以下叙述正确的是_____。
 - A) C 语句的结束符为分号
 - B) C 语句的结束符为逗号
 - C) C 语句的结束符为冒号
 - D) C 语句的结束符为句号
4. 以下叙述正确的是_____。
 - A) 在一个 C 语言程序中，有且只能有一个主函数
 - B) 在一个 C 语言程序中，可以有两个或两个以上的主函数
 - C) 在一个 C 语言程序中，主函数可以由用户来命名
 - D) 在一个 C 语言程序中，主函数名(main)后面的括号可以省略
5. 以下叙述正确的是_____。
 - A) 计算机的硬件系统可以直接识别并执行汇编语言程序

- B) 计算机的硬件系统可以直接识别并执行高级语言程序
- C) 计算机的硬件系统可以直接识别并执行机器语言程序
- D) 计算机的硬件系统可以直接识别并执行 C 语言程序

6. 以下叙述不正确的是_____。

- A) 在 C 语言程序中，主函数、子函数都可以由用户来命名
- B) 在 C 语言程序中，子函数可以由用户来命名，但主函数不能
- C) 在 C 语言程序中，主函数名只能用 main
- D) 在 C 语言程序中，主函数不一定要放在子函数的前面

二、问答题

1. 程序设计语言分为哪几种类型？
2. C 语言的主要特点有哪些？
3. C 语言程序由哪几个部分构成？
4. 写出 Visual C++ 6.0 环境下运行一个 C 语言程序的步骤。

三、程序设计题

1. 编写一个程序，输出字符串“C 语言为世界上应用最广泛的几种计算机语言之一”。
2. 参照本章内容，编写一个程序，求两个数之和。