

# 第一章 程序设计和C语言

---

作者：石璞东

参考资料：《C程序设计（第四版）》谭浩强

## 1.1 程序和指令

---

- 程序：所谓程序，就是一组计算机能识别和执行的指令；
- 指令：一个特定的指令序列，用来完成一定的功能，每一条指令使计算机执行特定的操作，只要让计算机执行这个程序，计算机就会**自动地**执行各条指令，有条不紊地进行工作；

## 1.2 计算机语言

---

### 1. 计算机编程语言的分类

- 机器语言
  - 概念：计算机能直接识别和接受的二进制代码成为机器指令，机器指令的集合就是机器语言，如10010011，我们可以将其理解为人类社会发展的原始阶段；
  - 操作方法：当编写完成用0和1组成的指令后，需要用纸带穿孔机以人工的方法在特制的黑色纸带上穿孔，在指定的位置上有孔代表1，无孔代表0，一个程序往往需要一卷长长的纸带，在需要运行此程序时，就将此纸带装在光电输入机上，当光电输入机从纸带读入信息时，有孔处产生一个电脉冲，指令变成电信号，让计算机执行各种操作；
  - 特点：
    - 优点：直接执行，速度快，资源占用少；
    - 缺点：可读性、可移植性差，编程复杂；
- 汇编语言
  - 概念：又称“符号语言”，即用一些英文字母和数字表示一个指令，如用 `ADD` 表示加法操作，用 `SUB` 表示减法操作，`LD` 表示“传送”，显然，计算机并不能直接识别和执行符号语言的指令，需要使用一种称为汇编程序的软件，将符号语言的指令转换为机器指令；
- 高级语言
  - 按转换方式分类：编译型语言、解释型语言；
  - 按客观系统描述分类：面向过程语言、面向对象语言；
  - 按编程范型分类：命令式语言、函数式语言、逻辑式语言、面向对象语言；

参考链接：<https://www.liuchuo.net/archives/130>

## 2. 高级语言的不同发展阶段

- 非结构化的语言：初期的语言属于非结构化的语言，编程风格比较随意，只要符合语法规则即可，没有严格的规范要求，程序中的流程可以随意跳转，人们往往追求程序执行的效率而采用了许多“小技巧”，使程序变的难以阅读和维护，早期的 BASIC、FORTRAN 和 ALGOL 等都属于非结构化的语言；
- 结构化语言：为了解决以上问题，提出了“结构化程序设计方法”，规定程序必须由具有良好特性的基本结构（顺序结构、分支结构、循环结构）构成，程序中的流程不允许随意跳转，程序总是由上而下顺序执行各个基本结构，C 语言属于结构化语言；
- 面向对象语言：近年来，在处理较大规模的问题时，开始使用面向对象的语言；

## 1.3 C语言介绍

### 1. 概述

C语言是国际上广泛流行的计算机高级语言，其祖先是 BCPL 语言（Basic Combined Programming Language）。1970年，美国 AT&T 贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，设计出了很多简单且很接近硬件的 B 语言（取 BCPL 的第一个字母），但 B 语言过于简单，功能有限；1972~1973年间，美国贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言，C 语言既保持了 BCPL 和 B 语言的优点（精炼，接近硬件），又克服了它们的缺点（过于简单，无数据类型等），C 语言的新特点主要表现在具有多种数据类型（如字符、数值、数组、结构体和指针等）。开发 C 语言的目的在于尽可能降低用它所写的软件对硬件平台的依赖程度，使之具有可移植性。

### 2. 主要特点

- 语言简洁、紧凑，使用方便、灵活；
- 运算符丰富（34种）；
- 数据类型丰富（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）；
- 具有结构化的控制语句（如 if...else 语句、while 语句、do...while 语句、switch 语句和 for 语句）；
- 语法限制不太严格，程序设计自由度大（如不对数组越界进行检查）；
- 允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作；
- 程序可移植性好；
- 生成目标代码的质量高，程序执行效率高；

## 1.4 案例展示

## 1. Let's say Hello,World!

```
1  #include <stdio.h>
2  int main() {
3      printf("Hello, World!\n");
4      return 0;
5  }
```

程序分析解释：

- 第一行：用来提供输入输出函数的相关信息，`stdio.h` 是系统提供的一个文件名，即 `standard input&output` 的缩写，文件后缀 `.h` 的意思是头文件（`header file`），因为这些文件都是放在程序各文件模块的开头的，输入输出函数的相关信息已事先放在 `stdio.h` 文件中，现在用 `#include` 指令把这些信息调入供使用；
- 第二行：其中 `main` 是函数的名字，表示主函数，`main` 前面的 `int` 表示此函数的类型是 `int` 类型，在执行主函数后会得到一个值（即函数值），其值为整型，每一个 C 语言程序都必须有一个 `main` 函数，函数体由花括号括起来；
- 第三行：这是一个输出语句，`printf` 是 C 编译系统提供的函数库中的输出函数，在上述代码中，`printf` 函数中双撇号内的字符串 `Hello,World!` 按原样输出，`\n` 是换行符，即在 `Hello,World!` 后，显示屏上的光标位置移到下一行的开头；
- 第四行：当 `main` 函数执行结束前将整数 `0` 作为函数值，返回到调用函数处；

为使程序规范和可移植性，一般建议将 `main` 函数指定为 `int` 型，并在 `main` 函数的最后加一个 `return 0;` 语句。

「注解」：C 语言中的两种注解方式：

- 单行注释：`//`；
- 块式注释：`/**/`；

## 2. 求两数之和

```
1  #include <stdio.h>
2  int main(){
3      int a,b,sum;
4      printf("请输入a、b的值: ");
5      scanf("%d %d",&a,&b);
6      sum = a + b;
7      printf("两数之和为: %d\n",sum);
8      return 0;
9  }
```

- `printf` 函数圆括号内有两个参数：一个是双撇号中的内容 `两数之和为: %d\n`，它是输出格式字符串，作用是输出用户希望输出的字符和输出的格式；第二个参数 `sum` 表示要输出的变量的值；

### 3. 求两数中的较大者

```
1  #include<stdio.h>
2  int main(){
3      int max(int x,int y);
4      int a,b,c;
5      printf("请输入a、b的值: ");
6      scanf("%d %d",&a,&b);
7      c = max(a,b);
8      printf("a和b中的较大者为:%d\n",c);
9      return 0;
10 }
11 int max(int x,int y){
12     int z;
13     x > y ? (z=x):(z=y);
14     return z;
15 }
```

- `scanf` 函数：其作用是输入变量 `a` 和 `b` 的值，其参数包括两部分内容，一是双撇号中的内容，它指定输入的数据按什么格式输入，`%d` 的含义是十进制整数形式；二是输入的数据准备放到哪里，即赋给哪个变量，如上述代码所示，`scanf` 函数中指定的是 `a` 和 `b`，在 `a` 和 `b` 的前面各有一个 `&`，在 `C` 语言中是 `&` 地址符，`&a` 的含义是变量 `a` 的地址，`&b` 是变量 `b` 的地址，执行 `scanf` 函数，从键盘读入两个整数，送到变量 `a` 和 `b` 的地址处，然后把这两个整数分别赋给变量 `a` 和 `b`。

### 2. C 语言程序的结构特点

- 一个程序由一个或多个源程序文件组成；
  - 一个源程序文件包含以下3部分：
    - 预处理指令
    - 全局声明
    - 函数定义
- 函数是 `C` 程序的主要组成部分；
- 一个函数主要包括两个部分；
  - 函数首部：`int max(int x, int y)`
    - 解释：第一个 `int` 表示函数类型，第二个 `int` 和第三个 `int` 表示参数类型，`max` 表示函数名，`x`、`y` 表示函数参数名；如果函数没有参数，可以写为 `int main(void)` 或 `int main()`；
  - 函数体：
    - 声明部分：定义在本函数中所用到的变量以及对本函数所调用函数的声明；
    - 执行部分：由若干个语句组成，指定在函数中所进行的操作；
- 程序总是从 `main` 函数开始执行的；
- 程序中对计算机的操作是由函数中的 `C` 语句完成的；

- 在每个数据声明和语句的最后必须有一个分号；
- c 语言本身不提供输入输出语句；
- 程序应当包含注释；

### 3. 如何运行 c 语言代码？

计算机不能直接识别和执行用高级语言写的指令，必须用编译程序把 c 源程序翻译成二进制形式的目标程序，然后在将该目标程序与系统的函数库以及其他目标程序连接起来，形成可执行的目标程序。

在编译好一个 c 源程序后，一般要经过以下几个步骤方可运行：

- 上机输入和编译源程序；
- 对源程序进行编译；
- 进行连接处理；
- 运行可执行程序，得到运行结果；



