

计算机导论与程序设计 [CS006001]

段江涛

机电工程学院



2020 年 9 月

lecture-3 主要内容

最简单的 C 语言程序设计—顺序程序设计

1 初识 C 语言程序 (讲解作业)

2 数据的输入输出

DevC++ 5.0 以前的版本

```
#include<stdio.h>                // standard input/output编译预处理指令
#include <stdlib.h>               // standard library function, eg. for
    system() 函数
int main()                       // 主函数
{
    printf("%d\n",p);
    system("pause");              // 窗口暂停, DevC++ 5.0以前的版本。[机试系统提
    交时, 一定注释或删除该语句]
    return 0;                    // 函数执行完毕返回函数值0
}                                 // 函数结束标志
```

求 5! 的 C 语言程序

```
#include<stdio.h>           // standard input/output编译预处理指令

int main()                   // 主函数
{                             // 函数开始标志

    int i,p;  // p表示被乘数, i表示乘数
    p=1;
    i=2;
    while(i<=5)
    {
        p=p*i;
        i++; // i = i + 1
    }
    printf("%d\n",p);

    return 0;                // 函数执行完毕返回函数值0
}                             // 函数结束标志
```

变量在使用之前首先要定义它的数据类型

```
#include<stdio.h>           // standard input/output编译预处理指令

int main()                   // 主函数
{                             // 函数开始标志

    int a,b;  // 定义变量a, b为整型数值, 同类型变量可以在一条语句中定义。
    float f;  // 定义变量f为单精度浮点数
    double d; // 定义变量d为双精度浮点数
    char c;   // 定义变量c为单个英文字母

    a=10;
    b=20;
    f=10.2;
    d=20.3;
    c='A';

    return 0;                // 函数执行完毕返回函数值0
}                             // 函数结束标志
```

if(条件表达式){ 表达式为真 (非 0) 时执行语句; }

```
#include<stdio.h>           // standard input/output编译预处理指令

int main()                   // 主函数
{                             // 函数开始标志

    int a=10;                // 定义变量a为整型数值, 定义变量时, 可以指定变量的初值
    if (a>=10)
    {
        printf("a>=10\n"); // \n为换行符
    }
    else
    {
        printf("a<10\n");  // \n为换行符
    }

    return 0;                // 函数执行完毕返回函数值0
}                             // 函数结束标志
```

while(条件表达式){ 表达式为真 (非 0) 时执行的语句;}

```
#include<stdio.h>           // standard input/output编译预处理指令
int main()                   // 主函数
{                             // 函数开始标志
    int a=10;                // 定义变量a为整型数值, 定义变量时, 可以指定变量的初值
    while(a>=0)
    {
        printf("a=%d\n",a); // \n为换行符
        a--; // a= a - 1
    }
    return 0;                // 函数执行完毕返回函数值0
}                             // 函数结束标志
```

常用格式描述符与数据类型的对应关系

格式符	对应的数据类型	备注
%d	int	
%f	float	
%c	char	
%lf	double	
%.2f	float	保留两位小数, 四舍五入。不适用于 scanf()。
%.2lf	double	保留两位小数, 四舍五入。不适用于 scanf()。
%x	int	十六进制显示
%ld	long int	

详见 p73, 表 3.6

输出语句 printf(“原样输出,% 格式符”, 对应变量的值);

```
#include<stdio.h>                // standard input/output编译预处理指令

int main()                        // 主函数
{                                  // 函数开始标志

    int a=10,b;    // 定义变量a, b为整型数值, 定义变量时, 可以指定变量的初值
    float f=10.2;  // 定义变量f为单精度浮点数
    double d; // 定义变量d为双精度浮点数
    char c;    // 定义变量c为单个英文字母
    f=10.2;
    d=20.356;
    c='A';
    printf("a=%d,b=%d,c=%c,f=%f,d=%.2lf\n",a,b,c,f,d); // %.2f, %.2lf保留
    两位小数
    return 0;                // 函数执行完毕返回函数值0
}                             // 函数结束标志
```

变量 b 没有被赋值, 将是一个随机值。

输入语句 scanf(“% 变量格式符”, & 变量名);

```
#include<stdio.h>           // standard input/output编译预处理指令

int main()                   // 主函数
{
    // 函数开始标志

    int a=10,b;              // 定义变量a, b为整型数值, 定义变量时, 可以指定变量的初值
    float f=10.2;           // 定义变量f为单精度浮点数
    double d; // 定义变量d为双精度浮点数
    char c='A';              // 定义变量c为单个英文字母, 字符输入以后讲
    printf("请输入整数和浮点数, 空格隔开:\n"); // 提示语句[可选]
    scanf("%d%f", &a, &f);    // 尽量简单, 不要有其它字符和'\n'
    printf("请输入两个浮点数, 空格隔开:\n"); // 提示语句[可选]
    scanf("%f%lf", &f, &d);
    printf("a=%d,b=%d,c=%c,f=%f,d=%lf\n", a,b,c,f,d); // \n为换行符
    return 0;                // 函数执行完毕返回函数值0
}                             // 函数结束标志
```

字符输出函数 putchar

```
#include<stdio.h>

int main()
{
    char a = 'B', b = 'O', c = 'Y'; //定义3个字符变量并初始化
    putchar(a); //向显示器输出字符B
    putchar(b); //向显示器输出字符O
    putchar(c); //向显示器输出字符Y
    putchar ('\n'); //向显示器输出一个换行符
    return 0;
}
```

字符输入函数 getchar, 遇到回车, 开始从缓冲区中接收字符。

```
#include<stdio.h>

int main()
{
    char a,b,c;    //定义字符变量a,b,c
    a = getchar(); //从键盘输入一个字符, 送给字符变量a
    b = getchar(); //从键盘输入一个字符, 送给字符变量b
    c = getchar(); //从键盘输入一个字符, 送给字符变量c
    putchar(a);    //将变量a的值输出
    putchar(b);    //将变量b的值输出
    putchar(c);    //将变量c的值输出
    printf("\na=%d,b=%d,c=%d,a=%c,b=%c,c=%c\n",a,b,c,a,b,c);
    return 0;
}
```

字符输入函数 `getchar`, 遇到回车, 开始从缓冲区中接收字符。

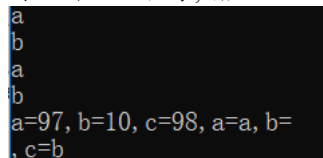
```
char a,b,c; //定义字符变量a,b,c
a = getchar(); //从键盘输入一个字符, 送给字符变量a
b = getchar(); //从键盘输入一个字符, 送给字符变量b
c = getchar(); //从键盘输入一个字符, 送给字符变量c
putchar(a); //将变量a的值输出
putchar(b); //将变量b的值输出
putchar(c); //将变量c的值输出
printf("\na=%d,b=%d,c=%d,a=%c,b=%c,c=%c\n",a,b,c,a,b,c);
```

从键盘输入 `abc` 回车, 观察结果, 应该是正确的结果。遇到回车, 开始从缓冲区中接收字符。

字符输入函数 `getchar`, 遇到回车, 开始从缓冲区中接收字符。

```
a = getchar(); //从键盘输入一个字符, 送给字符变量a='a'
b = getchar(); //从键盘输入一个字符, 送给字符变量b='\n'
c = getchar(); //从键盘输入一个字符, 送给字符变量c='b'
putchar(a); //将变量a的值输出a
putchar(b); //将变量b的值输出\n
putchar(c); //将变量c的值输出b
printf("\na=%d,b=%d,c=%d,a=%c,b=%c,c=%c\n",a,b,c,a,b,c);
```

再运行一次程序, 输入 a 回车, 输入 b 回车, 输入 c 回车, 观察结果。



```
a
b
a
b
a=97, b=10, c=98, a=a, b=
, c=b
```

开发平台上演示讲解

在开发平台, 以具体的示例, 详细讲解以下内容:

- int, float, double, char 数据类型, sizeof() 函数
- %d, %f, %c, %lf, %x 格式符的使用 (见 ppt 中的表格)
- if(){ }, while(){ } 简单语句
- char c; scanf(“%c”, &c); 接收输入的字符
- char c; c=getchar() 接收输入的字符, putchar() 输出一个字符
- 编程理解数字 ASCII 码与整数的对应关系以及大小写字符之间的关系。
- 避免数字, 字符在一条语句中输入的情况, 如:
scanf(“%d%c%d”,...);
- 重点理解字符缓冲区的概念, 以及消费无用字符的技巧。

欢迎批评指正！