

18-19学年秋季学期计算概论课后辅导讲义Lecture 2

陈婉雯

2018 年 10 月 12 日

1 本讲主题

- 常见的数据类型及存储格式
- 基于数据再认识printf()与scanf()
- 课堂程序精讲：简单的函数和数组

2 常见数据类型、存储和sizeof

2.1 存储

位(bit)：最小的存储单元（0或1） 字节(byte)：1字节为8位字：设计计算机时给定的自然存储单位，比如常说的32位、64位计算机，意思是一个字长为32位或64位。字长越大，说明数据转移越快，允许的内存访问更多。sizeof函数：以字节为单位。比如输入范例代码：

```
#include <stdio.h>
int main()
{
    printf("int: %d\n", sizeof(int));
    printf("long int: %d\n", sizeof(long));
    printf("short int: %d\n", sizeof(short));
    printf("long long int: %d\n", sizeof(long long int));
    printf("char: %d\n", sizeof(char));
    getchar();
    return 0;
}
```

注意这里用了个getchar函数来避免运行的时候程序一闪就结束了。输出为：

```
int: 4
long int: 4
short int: 2
long long int: 8
char: 1
```

输出结果可能会由于机器的不同而有所不同。大家也可以尝试一下查看其它类型的字节。

2.2 整数

整数就是没有小数部分的数。最常见的就是int类型。接下来的介绍也以int类型为主。int为有符号整型。一般来说，一个int要占用一个机器字长（通常为32位）。由于int是有符号的整型，为了表示负数，有一个位是符号位。ISO C 规定，int的范围为 $-32768 \sim 32767$ （16位，32768为2的15次方）。在声明整数为无符号类型时，它的范围为 $0 \sim 65536$ 。

在声明变量并初始化时，计算机会为变量创建、标记存储空间并赋值；在赋值运算符操作时，计算机会把值存在这个内存空间里。

3 再说printf()与scanf()

3.1 转换说明符

大部分同学在使用int类型时并没有遇到什么问题。但值得注意的是，当你想打印其他类型时，可能要注意说明符的使用。比如打印unsigned int 类型时要使用%u，打印long 类型要使用%ld。假如类型不符合时，可能会出现以下状况：

示例代码：

```
#include <stdio.h>
int main()
{
    int i=0;
    printf("i=%d\n",i);
    /* Can't pass the compilation if you don't initializing i! */
    i=-123;
    printf("i=%d\n",i);
    printf("i=%u\n",i);
    i=1000000000000;
    printf("i=%d\n",i);
    getchar();
}
```

输出结果：

```
i=0
i=-123
i=4294967173
i=1215752192
```

希望这个例子可以帮助同学们进一步了解计算机是怎么将10（高低电平）序列转换成我们能够理解的数据。在输入输出中，计算机会根据转换说明符，按照某种规则读取内存里的二进制存储数据。因此，如果我们用与数据不对应的转换说明符时，计算机可能会输出不符合要求的数据。转换说明有很多，你也可以要求计算机以八进制、十六进制、指数计数法等方式输出数据。具体方法可自行在网上查阅，或者阅读参考书。

注：如果在你的计算机上long和int的存储字节相同的话long和int可以混用（），但不推荐，因为不利于移植。

3.2 scanf()函数的坑

在我获得的作业反馈中，同学们遇到的大多数问题都是出现在用scanf()读取数据上。所以本节更详细地给大家讲解一下scanf()函数。scanf()也是用格式字符串表示字符输入的目标数据类型，参数列表是指向变量的指针。指针会在其后的课程再做讲解，现在可以简单理解为存储数据的地址。目前只需要记住两条简单规则：

- 如果用scanf读取基本变量类型的值，要在变量名前加上&（取地址符）
- 如果用scanf把字符串读入字符数组中，不要使用&。

scanf函数使用空白（换行符、制表符、空格）把输入分成多个字段，并且在把转换说明和字段匹配时会跳过空白。唯一的例外是用%c读取字符，此时scanf会读取每个字符，包括空白。

scanf如何读取整数：scanf函数每次读取一个字符，并且跳过所有空白字符，在遇到第一个非空白字符时才开始读取。因为要读取整数，所以scanf在找到一个数字或者正负号时就会保存这个字符，读取下一个字符。如果下一个字符是数字，它就会保存该数字并读取下一个字符，直至遇到非数字字符。在遇到一个非数字字符时，它便认为读到了整数的末尾。这个时候，它会把这个非数字字符放回输入，所以在下一次读取输入时，首先读到的是上一次读取丢弃的非数字字符。最后，scanf会计算已经读取的数字相应的数值并把值放入指定变量中。如果第一个非空白字符不是数字或者正负号时，scanf就会停在那里，并把这个字符放回输入中。在下一次读取输入时，又首先读到这个字符。所以如果程序只使用%d转换说明，scanf就一直无法越过该字符读下一个字符。

4 课堂程序精讲

4.1 简单函数

函数的好处在于可以减少书写重复代码，便于代码的维护和复用。举个例子（例程1-21）：这是一个判断输入整数是否为质数的函数，如果是质数则返回1，否则返回0。

```
int isPrime(int N){
    if(N<2) return 0;
    if(N==2) return 1;
    if(N%2==0) return 0;
    int i=2;
    while(i<N){
        if(N%i==0) return 0;
        i++;
    }
    return 1;
}
```

第一行的int是函数返回的数据的类型；isPrime是函数名，括号内的int N表示这个函数需要外界输入参数N，且N必须为整型。return后面的值则是函数的返回值。函数在执行了return后就会停止执行。

精讲：例程1-23.

函数原型：定义在开头/头文件里，告诉编译器函数的类型、接受的参数类型。

形式参数：函数定义的函数头中生命的变量，实际参数：出现在函数调用圆括号内的表达式。使用函数时会创建声明为形式参数的变量并初始化为实际参数的求值结果。（不改变实际参数）

4.2 简单数组

数组是按顺序储存的一系列类型相同的值。整个数组有一个数组名，通过整数下标访问数组中单独的项或元素。例如对于声明：

```
float a[20];
```

声明了a是一个有20个元素的数组，每个元素都可以储存float类型的值。

数组的声明、初始化方法有多种。例如：

```
int arrA[] = {12, 1, 65, -1, 98};
```

指定初始值，此时计算机将根据初始值自动开辟相应大小的存储空间，相当于int arrA[3]=12,1,65,-1,98;

```
int arrB[5] = {1, 2, 3};
```

声明时指定部分初始值;这个时候没有被指定初始值的会被设成0。相当于int arrB[5]=1,2,3,0,0;

```
int arrC[5];
```

声明时没有指定初值，此时必须指定数组大小，数组内的值没有初始化。

数据经常在循环中（尤其是for循环）使用。要注意的是，**数组元素编号从0开始**。使用数组时可以通过数组名和方括号运算符以及方括号中的整数值(称之为索引编号)读取或者设置数据成员的值。这让我们能够在循环中批量处理数据，极大地简化了程序。

精讲：例程1-33.为什么函数可以改变数组内的值？

5 作业选讲

思路、流程（画框图）、实际编写和debug