

第二十七节：整除求商。

【27.1 什么叫整除？】

最小的细分单位是“1”的除法运算就是整除，“1”不能再往下细分成小数点的除法运算就是整除。比如：

10 除以 4，商等于 2.5。-----（带小数点，这个不是整除）

10 除以 4，商等于 2，余数是 2。-----（这才是整除）

什么时候带小数点，什么时候是整除？取决于参与运算的变量类型。标准的 C 语言中，其实远远不止我前面所说的 unsigned char, unsigned int, unsigned long 这三种类型，比如还有一种叫浮点数类型的 float，当参与运算的变量存在 float 类型时，就可能存在小数点。关于小数点的问题以后再讲，现在暂时不深入讲解，现在要知道的是，unsigned char, unsigned int, unsigned long 这三种变量类型的除法都是属于整除运算，不带小数点的。

【27.2 整除的运算符号是什么样子的？】

10 除以 4，商等于 2，余数是 2，这个整除的过程诞生了两个结果，一个是商，一个是余数，与此对应，整除就诞生出两个运算符号，你如果想计算结果返回商就用“整除求商”的符号“/”，你如果想计算结果返回余数就用“整除求余”的符号“%”。咋一看，整除运算中用到的两个符号“/”和“%”都不是我们日常生活中熟悉的除号“÷”，我个人猜测是因为“÷”这个符号在电脑键盘上不方便直接输入，因此 C 语言的语法规则选用“/”和“%”作为整除的运算符号。

【27.3 整除求商“/”。】

整除求商的通用格式：

“保存变量” = “被除数” / “除数 1” / “除数 2” ... / “除数 N”；

跟之前讲的加减运算一样，赋值符号“=”左边的“保存变量”必须是变量，右边的可以是变量和常量的任意组合。如果右边只有两个参与运算的数据，就是整除求商的常见格式。

整除求商的常见格式：

“保存变量” = “被除数” / “除数”；

现在深入分析一下整除求商的运算规律。

（1）当除数等于 0 时。

我们都知道，数学运算的除数是不允许等于 0 的，如果在 51 单片机中非要让除数为 0，商会出现什么结果？我测试了一下，发现有一个规律：在 unsigned char 的变量类型下，如果“除数”是变量的 0，商等于十进制的 255（十六进制是 0xff）。如果“除数”是常量的 0，商等于十进制的 1。比如：

```
unsigned char a;
unsigned char b;
unsigned char y=0;
a=23/y; //除数变量 y 里面是 0，那么 a 的结果是 255(十六进制的 0xff)。
b=23/0; //除数是常量 0，那么 b 的结果是 1。
```

平时做项目要尽量避免“除数是 0”的情况，离它越远越好，但是既然除数不能为 0，为什么我非要做“除数为 0”时的实验呢？意义何在？这个实验的意义是，虽然我知道除数为 0 时会出错，但是我不知道这个错到底严不严重，会不会导致整个程序崩溃，当我做了这个实验后，我心中的石头才放下了，万一除数为

0 时，最多只是运算出错，但是不至于整个程序会崩溃，这样我心里就有了一个底，当哪天我某个程序崩溃跑飞时，我至少可以排除了“除数为 0”这种情况，引导我从其它方面去找 bug。

(2) 当被除数小于除数时。商等于 0。比如：

```
unsigned char c;  
c=7/10;    //c 的结果是 0。
```

(3) 当被除数等于除数时。商等于 1。比如：

```
unsigned char d;  
d=10/10;   //d 的结果是 1。
```

(4) 当被除数大于除数时。商大于 0。比如：

```
unsigned char e;  
unsigned char f;  
e=10/4;    //e 的结果是 2，大于 0。  
f=10/3;    //f 的结果是 3，大于 0。
```

【27.4 整除求商的自除简写。】

当被除数是“保存变量”时，存在自除运算的简写。

“保存变量” = “保存变量” / “除数” ；

上述自除运算的简写如下：

“保存变量” / = “除数” ；

比如：

```
unsigned char e;  
g/=5;    //相当于 g=g/5;
```

【27.5 整除求商有没有“自除 1”的特殊写法？】

加减法有自加 1 “++g” 和自减 1 “g--” 的特殊写法，但是除法不存在这种自除 1 的特殊写法，因为一个数除以 1 还是等于它本身，所以自除 1 没有任何意义，因此 C 语言语法中没有这种写法。

【27.6 整除求商的溢出。】

除法的溢出规律跟加法的溢出规律是一样的，所以不再多举例子。在实际项目中，为了避免一不小心就溢出的问题，我建议，不管加减乘除，凡是参与运算的变量全部都应该转化成 unsigned long 变量，转化的方法已经在前面章节讲过，不再重复讲解这方面的内容。

【27.7 例程练习和分析。】

现在编写一个程序来验证刚才讲到的整除求商：

程序代码如下：

```
/*---C 语言学习区域的开始。-----*/
```

```

void main() //主函数
{
    unsigned char a;
    unsigned char b;
    unsigned char c;
    unsigned char d;
    unsigned char e;
    unsigned char f;
    unsigned char g=10; //初始化为 10
    unsigned char y=0;  //除数变量初始化为 0。

    //（1）当除数等于 0 时。
    a=23/y;
    b=23/0; //这行代码在编译时会引起一条警告“Warning”，暂时不用管它。

    //（2）当被除数小于除数时。
    c=7/10;

    //（3）当被除数等于除数时。
    d=10/10;

    //（4）当被除数大于除数时。
    e=10/4;
    f=10/3;

    //（5）整除求商的简写。
    g/=5; //相当于 g=g/5;

    View(a);          //把第 1 个数 a 发送到电脑端的串口助手软件上观察。
    View(b);          //把第 2 个数 b 发送到电脑端的串口助手软件上观察。
    View(c);          //把第 3 个数 c 发送到电脑端的串口助手软件上观察。
    View(d);          //把第 4 个数 d 发送到电脑端的串口助手软件上观察。
    View(e);          //把第 5 个数 e 发送到电脑端的串口助手软件上观察。
    View(f);          //把第 6 个数 f 发送到电脑端的串口助手软件上观察。
    View(g);          //把第 7 个数 g 发送到电脑端的串口助手软件上观察。

    while(1)
    {
    }
}

/*---C 语言学习区域的结束。-----*/

```

在电脑串口助手软件上观察到的程序执行现象如下：

开始...

第 1 个数

十进制:255

十六进制:FF

二进制:11111111

第 2 个数

十进制:1

十六进制:1

二进制:1

第 3 个数

十进制:0

十六进制:0

二进制:0

第 4 个数

十进制:1

十六进制:1

二进制:1

第 5 个数

十进制:2

十六进制:2

二进制:10

第 6 个数

十进制:3

十六进制:3

二进制:11

第 7 个数

十进制:2

十六进制:2

二进制:10

分析：

通过实验结果，发现在单片机上的计算结果和我们的分析是一致的。

【27.8 如何在单片机上练习本章节 C 语言程序？】

直接复制前面章节中第十一节的模板程序，练习代码时只需要更改“C 语言学习区域”的代码就可以了，其它部分的代码不要动。编译后，把程序下载进带串口的 51 学习板，通过电脑端的串口助手软件就可以观察到不同的变量数值，详细方法请看第十一节内容。