

第六十九节： 宏函数 sizeof()。

【69.1 宏函数 sizeof() 的基础知识。】

宏函数 sizeof() 是用来获取某个对象所占用的字节数。既然是“宏”，就说明它不是单片机执行的函数，而是单片机之外的 C 编译器执行的函数（像#define 这类宏语句一样），也就是说，在单片机上电之前，C 编译器在电脑端翻译我们的 C 语言程序的时候，一旦发现了这个宏函数 sizeof，它就会在电脑端根据 C 语言程序的一些关键字符（比如“unsigned char, [,]”这类字符）来自动计算这个对象所占用的字节数，然后再把我们 C 语言程序里所有的 sizeof 字符替换等效成一个“常量数字”，1 代表 1 个字节，5 代表 5 个字节，1000 代表 1000 个字节。所谓在单片机之外执行的宏函数，就是说，在“计算”这些对象所占的字节数的时候，这个“计算”的工作只占用电脑的内存（C 编译器是在电脑上运行的），并不占用单片机的 ROM 容量和内存。而其它在单片机端执行的“非宏”函数，是占用单片机的 ROM 容量和内存。比如：

```
unsigned char a;           //变量。占用 1 个字节
unsigned int b;            //变量。占用 2 个字节
unsigned long c;          //变量。占用 4 个字节
code unsigned char d[9];   //常量。占用 9 个字节

unsigned int Gul6GetBytes; //这个变量用来获取字节数

Gul6GetBytes=sizeof(a);    //单片机上电后，在单片机程序里等效于 Gul6GetBytes=1;
Gul6GetBytes=sizeof(b);    //单片机上电后，在单片机程序里等效于 Gul6GetBytes=2;
Gul6GetBytes=sizeof(c);    //单片机上电后，在单片机程序里等效于 Gul6GetBytes=4;
Gul6GetBytes=sizeof(d);    //单片机上电后，在单片机程序里等效于 Gul6GetBytes=9;
```

上述的“sizeof 字符”在进入单片机的层面的时候，已经被编译器预先替换成对应的“常量数字”的，这个“常量数字”就代表所占用的字节数。

【69.2 宏函数 sizeof() 的作用。】

在项目中，通常用在两个方面：一方面是用在求一个数组的大小尺寸，另一方面是用在计算内存分配时候的偏移量。当然，sizeof 并不是“刚需”，如果没有 sizeof 宏函数，我们也可以人工计算出一个对象所占用的字节数，只是，人工计算，一方面容易出错，另一方面代码往往“动一发而牵全身”，改一个变量往往就会涉及很多地方需要配合调整更改，没法做到“自由裁剪”的境界。下面举一个程序例子：要把 3 个不同长度的数组“合并”成 1 个数组。

第一种情况：在没有使用 sizeof 宏函数时，人工计算字节数和偏移量：

```
unsigned char a[2]={1,2}; //占用 2 个字节
unsigned char b[3]={3,4,5}; //占用 3 个字节
unsigned char c[4]={6,7,8,9}; //占用 4 个字节
unsigned char HeBing[9]; //合并 a,b,c 在一起的数组。这里的 9 是人工计算 a,b,c 容量累加所得。
unsigned char i; //循环变量 i

for(i=0;i<2;i++) //这里的 2，是人工计算出 a 占用 2 个字节
```

```

{
    HeBing[i+0]=a[i]; //从 HeBing 数组的偏移量第 0 个地址开始存放。
}

for(i=0;i<3;i++) //这里的 3，是人工计算出 b 占用 3 个字节
{
    HeBing[i+2]=b[i]; //这里的 2 是人工计算出的偏移量。a 占用了数组 2 个字节。
}

for(i=0;i<4;i++) //这里的 4，是人工计算出 c 占用 4 个字节
{
    HeBing[i+2+3]=c[i]; //这里的 2 和 3 是人工计算出的偏移量，a 和 b 占用了数组 2+3 个字节。
}

```

第二种情况：在使用 sizeof 宏函数时，利用 C 编译器自动来计算字节数和偏移量：

```

unsigned char a[2]={1,2}; //占用 2 个字节
unsigned char b[3]={3,4,5}; //占用 3 个字节
unsigned char c[4]={6,7,8,9}; //占用 4 个字节
unsigned char HeBing[sizeof(a)+sizeof(b)+sizeof(c)]; //C 编译器自动计算字节数
unsigned char i;

for(i=0;i<sizeof(a);i++) //C 编译器自动计算字节数
{
    HeBing[i+0]=a[i];
}

for(i=0;i<sizeof(b);i++) //C 编译器自动计算字节数
{
    HeBing[i+sizeof(a)]=b[i]; //C 编译器自动计算偏移量
}

for(i=0;i<sizeof(c);i++) //C 编译器自动计算字节数
{
    HeBing[i+sizeof(a)+sizeof(b)]=c[i]; //C 编译器自动计算偏移量
}

```

【69.3 例程练习和分析。】

现在编写一个练习的程序：

```

/*---C 语言学习区域的开始。-----*/

unsigned char a[2]={1,2}; //占用 2 个字节

```

```

unsigned char b[3]={3,4,5}; //占用 3 个字节
unsigned char c[4]={6,7,8,9}; //占用 4 个字节
unsigned char  HeBing[sizeof(a)+sizeof(b)+sizeof(c)];//C 编译器自动计算字节数
unsigned char i;

void main() //主函数
{
    for(i=0;i<sizeof(a);i++)  //C 编译器自动计算字节数
    {
        HeBing[i+0]=a[i];
    }

    for(i=0;i<sizeof(b);i++)  //C 编译器自动计算字节数
    {
        HeBing[i+sizeof(a)]=b[i];  //C 编译器自动计算偏移量
    }

    for(i=0;i<sizeof(c);i++)  //C 编译器自动计算字节数
    {
        HeBing[i+sizeof(a)+sizeof(b)]=c[i]; //C 编译器自动计算偏移量
    }

    for(i=0;i<sizeof(HeBing);i++)  //利用宏 sizeof 计算出 HeBing 数组所占用的字节数
    {
        View(HeBing[i]);  //把 HeBing 所有数据挨个依次全部发送到电脑端观察
    }

    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:1

十六进制:1

二进制:1

第 2 个数

十进制:2

十六进制:2

二进制:10

第 3 个数

十进制:3

十六进制:3

二进制:11

第 4 个数

十进制:4

十六进制:4

二进制:100

第 5 个数

十进制:5

十六进制:5

二进制:101

第 6 个数

十进制:6

十六进制:6

二进制:110

第 7 个数

十进制:7

十六进制:7

二进制:111

第 8 个数

十进制:8

十六进制:8

二进制:1000

第 9 个数

十进制:9

十六进制:9

二进制:1001

分析:

HeBing[0]为 1。

HeBing[1]为 2。

HeBing[2]为 3。

HeBing[3]为 4。

HeBing[4]为 5。

HeBing[5]为 6。

HeBing[6]为 7。

HeBing[7]为 8。

HeBing[8]为 9。

【69.4 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。