

### 第三十三节：逻辑运算符的“按位取反”和“非”运算。

#### 【前面章节漏讲的补充。】

前面的章节中，漏讲了“与，或，异或”的简写格式，在这类运算中，当赋值语句左边的“保存变量”也是参与运算的变量本身时，存在简写的语法格式，比如：

```
a&=0x01; //相当于 a=a&0x01;
a|=0x01; //相当于 a=a|0x01;
a^=0x01; //相当于 a=a^0x01;
```

#### 【33.1 “按位取反”运算。】

“按位取反”运算也是以位为单位进行运算的。位是指二进制中的某一位，位只能是 0 或者 1。跟前面“加、减、乘、除、与、或、异或”有点不一样的地方是，“按位取反”的运算只有 1 个对象，它不像加法运算那样可以与其它第 2 个对象产生关系，比如“a 加 b”这里有 2 个对象 a 和 b，而“a 按位取反”只有 1 个对象 a。一个数的“按位取反”运算就是把该数转换成二进制后对每一位的“取反”运算。

“按位取反”运算的符号是波浪符号“~”。运算规律是：针对一个数的“按位取反”，先将其展开成二进制的格式，然后每个位取反，所谓取反就是 1 的变成 0，0 的变成 1。

现在举一个完整的例子来分析“~”运算的规律。有两个 unsigned char 类型的十进制数分别是 5 和 0，求~5 和~0 的结果分别是多少？分析步骤如下：

第一步：先把参与运算的两个数以二进制的格式展开。十进制转二进制的方法请参考前面第 14, 15, 16 节的内容。

```
十进制 5 的二进制格式是： 00000101。
十进制 0 的二进制格式是： 00000000。
```

第二步：将它们二进制格式的每一位取反，1 的变成 0，0 的变成 1。

(a) 对 5 的按位取反。

```
十进制的 5          ->  ~00000101
“按位取反”运算结果是 ->  11111010
```

(b) 对 0 的按位取反。

```
十进制的 0          ->  ~00000000
“按位取反”运算结果是 ->  11111111
```

第三步：

(a) 把二进制的 11111010 转换成十六进制是：0xFA。转换成十进制是 250。所以~5 的结果是 250。

(b) 把二进制的 11111111 转换成十六进制是：0xFF。转换成十进制是 255。所以~0 的结果是 255。

#### 【33.2 “非”运算。】

注意，“非”运算不是以位为单位进行运算的。“非”跟“按位取反”有点相似，但是区别也明显。“按位取反”是以位为单位进行运算的，侧重在局部。而“非”是针对一个数的整体，侧重在全局。“非”只有

两种状态“假”和“真”。0 代表假，大于 0 的数值代表真，也可以说“非”假即真，“非”真即假。不是假的就是真的，不是真的就是假的。强调的是两种状态的切换。在数值表示上，用 0 代表假的状态，用 1 代表真的状态。“非”的对象也只有 1 个，它不像加法运算那样可以与其它第 2 个对象产生关系，比如“a 加 b”这里有 2 个对象 a 和 b，而“a 的非”只有 1 个对象 a。

“非”运算的符号是感叹号“!”，注意输入这类运算符号的时候不能用汉字输入法，而是要切换到英文字符的输入法下再输入，否则编译不通过（其它运算符也一样，都要求在字符输入法下输入）。“非”运算的规律是：针对某个数的“非”，不管此数有多大，只要它大于 0，那么被“非”后就一定是 0。也不管此数是什么变量类型，只要它数值等于 0，那么被“非”后就一定是 1，而不是 0xff 或者 0xffff 之类。

现在举一个完整的例子来分析“!”运算的规律。有两个 unsigned char 类型的十进制数分别是 5 和 0，求!5 和!0 的结果分别是多少？分析思路如下：

(a) 针对 5 的“非”运算。

5 大于 0，是一个整体，被“非”后为 0。

(b) 针对 0 的“非”运算。

0 就是 0，是一个整体，被“非”后为 1。

### 【33.3 例程练习和分析。】

现在编写一个程序来验证刚才讲到的“按位取反”和“非”运算：

程序代码如下：

```
/*---C 语言学习区域的开始。-----*/

void main() //主函数
{
    unsigned char a=5;
    unsigned char b=5;
    unsigned char c=0;
    unsigned char d=0;

    a=~a;
    b=!b;

    c=~c;
    d=!d;

    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。

    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:250

十六进制:FA

二进制:11111010

第 2 个数

十进制:0

十六进制:0

二进制:0

第 3 个数

十进制:255

十六进制:FF

二进制:11111111

第 4 个数

十进制:1

十六进制:1

二进制:1

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。

### 【33.4 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。