

第三十一节：逻辑运算符的“或”运算。

【31.1 “或”运算。】

“或”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。两个数的“或”运算就是转换成二进制后每一位的“或”运算。

“或”运算的符号是“|”。运算规律是：两个位的“或”运算，如果两个位都是0，那么运算结果才是0，否则只要其中有一位是1，那么运算结果必定是1。比如：

```
0|0 等于 0。
0|1 等于 1。
1|0 等于 1。
1|1 等于 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 12 和 9，求 12|9 的结果是多少？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 13, 14, 15 节的内容。

```
十进制 12 的二进制格式是：00001100。
十进制 9 的二进制格式是： 00001001。
```

第二步：二进制数右对齐，按上下每一位进行“或”运算。

```
十进制的 12    ->   00001100
十进制的 9     ->   |00001001
“或”运算结果是 ->   00001101
```

第三步：把二进制的 00001101 转换成十六进制是：0x0D。转换成十进制是 13。所以 12|9 的结果是 13。

上一节讲的“与”运算最常见的用途是可以指定一个变量的某位清 0，而本节的“或”运算刚好相反，“或”运算最常见的用途是可以指定一个变量的某位置 1，其它位保持不变。比如一个 unsigned char 类型的变量 b，数据长度一共是 8 位，从右往左：

```
想让第 0 位置 1，其它位保持不变，只需跟十六进制的 0x01 相“或”：b=b|0x01。
想让第 1 位置 1，其它位保持不变，只需跟十六进制的 0x02 相“或”：b=b|0x02。
想让第 2 位置 1，其它位保持不变，只需跟十六进制的 0x04 相“或”：b=b|0x04。
想让第 3 位置 1，其它位保持不变，只需跟十六进制的 0x08 相“或”：b=b|0x08。
想让第 4 位置 1，其它位保持不变，只需跟十六进制的 0x10 相“或”：b=b|0x10。
想让第 5 位置 1，其它位保持不变，只需跟十六进制的 0x20 相“或”：b=b|0x20。
想让第 6 位置 1，其它位保持不变，只需跟十六进制的 0x40 相“或”：b=b|0x40。
想让第 7 位置 1，其它位保持不变，只需跟十六进制的 0x80 相“或”：b=b|0x80。
```

根据上述规律，假设 b 原来等于十进制的 84（十六进制是 0x54，二进制是 01010100），要想把此数据的第 0 位置 1，只需 b=b|0x01。最终 b 的运算结果是十进制是 85（十六进制是 0x55，二进制是 01010101）。把它们展开成二进制格式的运算过程如下：

```
十进制的 84    ->   01010100
十六进制的 0x01 ->   |00000001
“或”运算结果是 ->   01010101
```

【31.2 例程练习和分析。】

现在编写一个程序来验证刚才讲到的“或”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;
    unsigned char b=84; //十六进制是 0x54，二进制是 01010100。

    a=12|9;
    b=b|0x01;

    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。

    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:13
十六进制:D
二进制:1101

第 2 个数
十进制:85
十六进制:55
二进制:1010101
```

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。

【31.3 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。