

C 语言

第 15 讲、位操作

张晓平

武汉大学数学与统计学院

2017 年 7 月 8 日

1. 位操作介绍

1. 位操作介绍



C 语言的设计具备了汇编语言的运算能力，它支持全部的位操作符。位操作符是对字节或字中的位进行测试、置位或移位处理，在对微处理器的编程中，特别适合对寄存器、I/O 端口进行操作。

C 语言提供了六种位操作运算符：

- ▶ 位与
- ▶ 位或
- ▶ 位异或
- ▶ 取反
- ▶ 左移
- ▶ 右移

这些运算符只能用于整型操作数，即只能用于带符号或无符号的 `char`, `short`, `int` 与 `long` 类型。

&	位与	如果两个相应的二进制位都为 1，则该位的结果值为 1，否则为 0
	位或	两个相应的二进制位中只要有一个为 1，该位的结果值为 1
^	位异或	若参加运算的两个二进制位值相同则为 0，否则为 1
~	取反	一元运算符，用来对一个二进制数按位取反，即将 0 变 1，将 1 变 0
«	左移	用来将一个数的各二进制位全部左移 N 位，右补 0
»	右移	将一个数的各二进制位右移 N 位，移到右端的低位被舍弃，对于无符号数，高位补 0

定义 (位与运算) 参加运算的两个数据，按二进制位进行“与”运算。如果两个相应的二进制位都为 1，则该位的结果值为 1；否则为 0。

例 计算 $3 \& 5$

解 因 $3 = 00000011(2)$, $5 = 00000101(2)$, 而位与运算
为

$$\begin{array}{r} 00000011 \\ \& 00000101 \\ \hline 00000001 \end{array}$$

故 $3 \& 5 = 1$ 。


```
1 // and1.c:
2 #include <stdio.h>
3 int main(void)
4 {
5     int a=3;
6     int b = 5;
7     printf("%d & %d = %d\n", a, b, a&b);
8     return 0;
9 }
```

位与的用途：

► 清零

若想对一个存储单元清零，即使其全部二进制位为 0，只要找一个二进制数，其中各位符合以下条件：原来的数中为 1 的位，新数中相应位为 0。然后使二者进行 & 运算，即可达到清零目的。

例 设原数为 $43 = 00101011(2)$ ，另找一个数 $148 = 10010100(2)$ ，将两者做位与运算：

```
    00101011
&   10010100
-----
    00000000
```

```
1 // and2.c:
2 #include <stdio.h>
3 int main(void)
4 {
5     int a = 43;
6     int b = 148;
7     printf("%d & %d = %d\n", a, b, a&b);
8     return 0;
9 }
```

► 取一个数中某些指定位

若有一个整数 `a` (2 byte), 想要取其中的低字节, 只需要将 `a` 与 8 个 1 做位与运算即可。

例

```
      00101100 10101100
&    00000000 11111111
-----
      00000000 10101100
```

► 保留指定位

例 有一个数 $84 = 01010100(2)$ ，想把其中从左边算起的第 3, 4, 5, 7, 8 位保留下来，可做以下运算：

```
    01010100
&   00111011
-----
    00010000
```

一元运算符~ 将每个 1 变为 0，将每个 0 变为 1。

例

```
~  00111011
-----
   11000100
```