



链滴

位运算

作者: [xiejicheng](#)

原文链接: <https://ld246.com/article/1566631178084>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

位运算

程序中的数在计算机内存中是以二进制形式存在，位运算就是直接对整数在内存中对应的二进制进行作。

二进制

十进制转二进制：除2取余，逆序排列。

补码

数值有正负之分，二进制只有0和1，需要设定方法来代表正负。

原码

设定：二进制中最高位为0代表正，为1表示负。

例：0000 0110 对应的十进制为6，而1000 0110 对应的十进制为-6。这种表示被称为 **原码**。

原码存在问题，执行运算时可能会出现错误，例如1+ (-2) 的二进制运算：

```
0000 0001 + 1000 0010
= 1000 0011
= -3
```

问题就出在代表正负的最高位。

反码

设定：二进制各位置的0和1互换，例如 0000 0110 的反码为 1111 1001。

运算前先反码，得到结果后再次反码回来。

```
000 0001 + 1111 1101
=1111 1110
#转换成十进制前，再次反码
=-1
```

但是也存在问题，如1+(-1)：

```
0000 0001 + 1111 1110
= 1111 1111
=1000 0000
=-0
```

补码

补码是为了让负数变成能够加的正数，也就是 **负数的补码 = 负数的绝对值取反 + 1**，例如-1的补码：

```
-1 的绝对值 1
= 0000 0001 # 1的二进制原码
= 1111 1110 # 原码取反
```

= 1111 1111 # +1 后得到补码