

## 关于~取反的问题

笔记本： 我的第一个笔记本

创建时间： 2020/8/31 12:56

更新时间： 2020/8/31 15:15

作者： 1667567382@qq.com

---

## 关于~取反的问题

取反，很容易理解，对于这个数的二进制按位取反，0变成1，1变成0就好。

几个知识点记录一下吧。

1. 计算机中数值是按照补码的方式存储的，
2. 正数的补码是自己，
3. 负数的补码是除符号位取反最后加1。
4. 由补码求原码那也是差不多了，正数的话都一样，负数的话就要减掉1，再除符号位取反。

### 那么遇到的问题是什么呢？

现在需要对一个int值进行取反，

5的取反得到的结果为-6

0x00000005---取反--->0xffffffffA

那么现在看起来结果好像是这个，但是这个**运算出来的结果，计算机认为是补码**，那么实际上的结果原码是多少呢？

**0xffffffffA--减去1-->0xffffffff9---除符号位取反，这里的取反和~不一样，要除去符号位----->0x80000006，所以最后得到的结果是-6.**

---

实际开发中遇到的问题是，在数据传输的过程中，我们拿到了short类型的两个字节，但是，拿到的是个字符串，而且直接就转成了int类型，所以需要对这个进行处理，如果是正数那当然没问题，正数的原码和补码都是一样的，直接当初原码处理好了，但是是负数的时候，传过来的0101就是补码了，我们现在就是直接拿这个补码当成原码在处理了，这个值直接就变成了正数了，因为强转的时候都在低位了。

那么需要对其转化，高位字节还要左移8位，才是对的。

首先看低位的8位，int的32位，最后8位就是数据。那么转成原码需要减去1，然后取反。首先减去1就要大于0，不能等于0，其次取反的时候是对32位全部取反了，那么我们对不必要的前24位需要处理一下，让其全为1的话，取反就是0了，正负关系可以自己乘个-1来保证。后面的8位就是要的数据，可以0xffffffff左移8位就是0xfffff00这个数，再将其与源数据|操作，就可以得到想要的结构，0xfffff--，--为我们的数据。然后再取反就可以了。

然后高位的8位还比较麻烦，麻烦在于0xffffffff左移16位以后|操作以后，对于最后的8位会是0，而不是1，所以可以再进行|这个数0x000000ff，就可以得到那种除了那有数据的8位，其他全是1，然后再取反就可以得到正确的数据了。

