**C语言源码反码补码**

**1、数据在内存中存储的时候都是以二进制的形式存储的.**

int num = 10;

原码、反码、补码都是二进制.只不过是二进制的不同的表现形式.

**数据是以补码的二进制存储的.**

**2、数据正负性**

为了可以表示正负性.使用最高为来表示这个数的正负性.

如果最高为是0 那么表示这个数是1个正数

如果最高为是1 那么表示这个数是1个负数.

所以,来表示数据的只有31位. 所以,1个int类型的变量.

最小值是: -2147483648 最大值是:2147483647

**3、原码**

最高位表示符号位. 剩下的位数.是这个数的绝对值的二进制.

绝对值: 正数的绝对值是自己,负数的绝对值去掉负号.

**4、反码**

正数的反码就是其原码.

负数的反码就是在其原码的基础之上 符号位不变,其他位取反.

**5、补码**

正数的补码就是其原码.

负数的补码就是在其反码的基础之上+1

10.

10的原码:00000000 00000000 00000000 00001010

10的反码:00000000 00000000 00000000 00001010

10的补码:00000000 00000000 00000000 00001010

-8

-8的原码:10000000 00000000 00000000 00001000

-8的反码:11111111 11111111 11111111 11110111

-8的补码:11111111 11111111 11111111 11111000

**6、任何数据都是以其二进制的补码形式存储在内存中的.**

**7、什么数据要以补码的形式存储呢?**

* 使用补码，可以将符号位和其它位统一处理；
* 同时，减法也可按加法来处理。

计算机中只有加法没有减法.为了更加低成本的计算出结果,所以使用补码来存储数据.

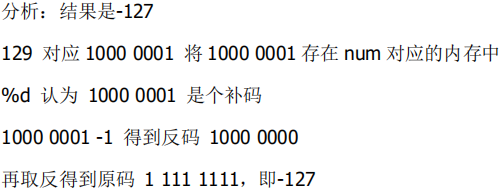
如3 - 2;

这个减法运算对于计算机而言它的理解是 3 + (-2); 1

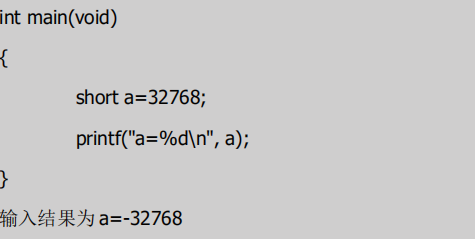
**8、常见例子**

**例子1：**

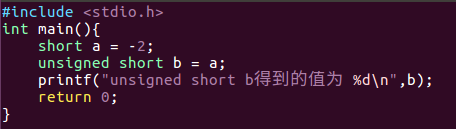




**例子2：**



**例子3：**





在unsigned中原码不分正负号

以int为例，原本int有32位bit。

* 如若没有unsigned， int 的第一位要表示正负，也就是说实际上int只有31位可以储存数据
* 可是若有unsigned，int的数据统一为正。Int也就不用再分正负号了，整个32位都可以储存数据！！并且在unsigned类型下，原码就是补码。