火星时代教育

位运算

—虚幻4高级程序开发专业—

前言

首先，在计算机中，所有的数都是以二进制的方式存在的！**位运算**主要是**针对数据以二进制的方式进行运算**，具有操作高效的特性。但是同时，由于二进制在程序中编码中很难展示，也导致位运算在当下编程中被使用的越来越少（框架级设计中很常见，产品开发中不常见）。位运算在一些特定情况下可以帮助我们解决一些非常棘手的问题，从而达到高效和节省内存的目的。

注意：位运算的规则不受语言约束，在常规的编程语言中规则相同，我们的课程只针对C++，所以内容呈现以C++为主。

PART 1

运算符介绍

Operator

运算符

参与位运算的运算符有按位与，按位或，按位异或，按位取反，左移，右移，计算符号参照下表

切记，进行位运算时，如果要做推断校验，需要先把数据转换为二进制，再进行运算

符号	记做	描述
&	按位与	a&b
|	按位或	a|b
^	按位异或	a^b
~	按位取反	~a
<<	左移	a<<b
>>	右移	a>>b

运算法

当两个数做位运算时，计算机将两个数转为二进制编码，然后在最低位进行对齐，然后进行计算。参照下图

注意，高位不够可以用零填补做参考，例如5对齐后最高位用零填补

运算数（十进制）	二进制
5	0101
13	1101
按位与&（5）	0101

按位与

运算规则：相同位的两个数字都为1，则为1；有一为0即为0。

运算数（十进制）	二进制
5	0101
13	1101
按位与&（5）	0101

按位或

运算规则：相同位数据，有一为1，即为1

运算数（十进制）	二进制
5	0101
13	1101
按位与|（13）	1101

按位异或

运算规则：相同位数据，不同为1，相同为0

运算数（十进制）	二进制
5	0101
13	1101
按位与^（8）	1000

按位取反

运算规则：数据位为0则变为1，为1则变为0，需要将高位用零补齐后再进行转换

注意：5取反后二进制是1010，那么1010（补码）转换为原码就是-6

如果被转换的数据是无符号的，需要考虑符号位不参与转换

运算数（十进制）	二进制
5	0101
按位取反~（-6）	1010

按位左移 <<

运算规则：将数据位向左移动，新进位用0填补。

注意，左移相当于乘以2的移动数的次方

运算数（十进制）	二进制
10	0000 1010
2	
左移<<（40）	0010 1000

运算数（十进制）	二进制（补码）
-10	1111 0110
2	
左移<<（-40）	1101 1000

按位右移 >>

运算规则：正数右移，将数据位向右移动即可。负数右移，将数据位向右移动，缺失位用1填补

注意，正数右移相当于除以2的移动数的次方

运算数（十进制）	二进制
10	0000 1010
2	
右移>>（2）	0000 0010

运算数（十进制）	二进制（补码）
-10	1111 0110
2	
右移>>（-3）	1111 1101

PART 2

优先级

Priority

优先级

按位运算符也存在优先级，可参照下表。如果拿不准优先级，建议使用括号提升优先级

排序	符号
1	~
2	<<, >>
3	&
4	^
5	|
6	&=, ^=, |=, <<=, >>=

PART 3

常规使用

Common use

快速乘除2

正数的右移是除以2的次方。正负数的左移是乘以2

加密解密一个数字

按位异或运算存在两次异或同一个数，将再次获得原有数据，一般我们可以用来进行数据加密。即双方共同约定一个密钥码，发送消息前将数据异或密钥码，接收方收到消息后使用相同的密钥码进行按位异或，将得到发送前的数据

```
int A = 503;

int B = 423;

int C = A ^ B;

int D = C ^ B;//D将等于A
```

交换AB两个变量值

借助按位异或运算，可以快速将两个数进行交换，而不通过第三个变量。

```
int A = 503;

int B = 423;

A = A ^ B;

B = A ^ B;

A = A ^ B;
```

获取二进制最后一位值

借助按位与即可完成，如下

```
int A = 664;

int B = A & 0x1;
```

快速判断奇偶数

奇数和偶数的特点就是是否能被2整除，也可以理解是否是2的整倍数。那么我么可以得知奇数的二进制最后一位肯定是1（如果为0说明是正好满2进一），所以我们可以通过判断一个数的二进制最后一位是否是0或是1，来裁定是否是偶数或是奇数。

```
int A = 664;

bool bEven = A & 0x1 == 0;
```

取二进制的末尾3位或4位

```
int A = 664;

int B = A & 7;//后三位

int C = A & 0xf;//后四位
```

获取二进制第N位上的数

```
int A = 664;

int N = 7;

int B = A >> （N - 1）& 0x1;//A转换到二进制后，B是第7位上的数字
```

把第N位变为1或0

```
int A = 664;

int N = 5;

int B = 1 << （N - 1）| A;//将第5位上的数字改为1

int C =~(1 <<（N - 1））& A;//将第5位上的数字改为0
```

THANK YOU

火星时代 虚幻4高级程序