**2021/03/12_x86逆向_第7课_MagicNumber的推导、取模运算**

**笔记本:** x86逆向-C
**创建时间:** 2021/3/12 星期五 10:03
**作者:** ileemi

---


- 取模
  - 模除数为常量且为正数
  - 模除数为常量且为负数 jns、 abs


# 取模

## 模除数为常量且为正数

模除为常量（取模符右操作数）时，编译器没有对其进行优化，代码示例如下：

```
printf("%d\n", 10 % argc);
// 对应的汇编代码
mov eax, 0Ah
cdq
idiv [esp+argc]
push edx
```

> 模除为常量且不是2的幂时，VC++6.0编译器，同样没有优化，但是高版本的VS编译器对其进行了优化，代码示例：

- VC++6.0

```
printf("%d\n", argc % 7);
// 对应的汇编代码
mov eax, [esp+argc]
mov ecx, 7
cdq
idiv ecx
push edx
```

- VS 2019：

```
printf("%d\n", argc % 7); // 优化方法：r = a - qb
// 对应的汇编代码
mov esi, [ebp+argc]
```

```
mov eax, 92492493h
imul esi
add edx, esi
sar edx, 2
mov ecx, edx
shr ecx, 1Fh
add ecx, edx // ecx = esi / 7
lea eax, ds:0[ecx*8]
sub eax, ecx // eax = ecx * 7
sub esi, eax // esi = esi - ecx * 7
push esi
```

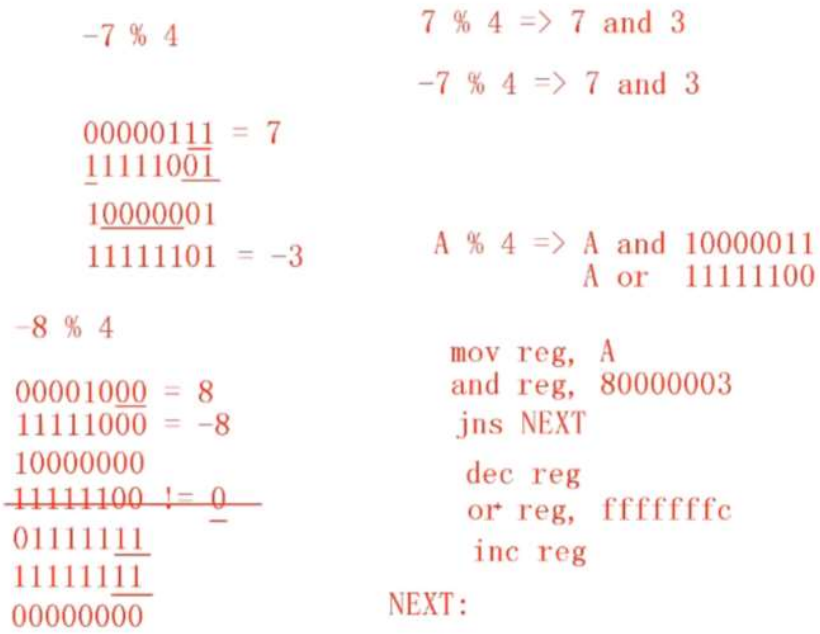
无符号被模数 模 2以及2的幂时，没有必要使用除法指令，可以判断奇偶性，做位运算，取对应位数（2^N，取N位），代码示例：

```
printf("%d\n", (unsigned int)argc % 8);
mov eax, [esp+argc]
and eax, 7
push eax
```

## 模除数为常量且为负数 jns、 abs

有符号被模数 模 -2以及-2的幂时，会多一个 "jns" 的跳转指令，是一个代码定式 "**and jns dec or inc**"。经过测试，VC++6.0 和 VS2019 的优化方案基本基址，代码示例：

```
printf("%d\n", argc % -8);
// 对应的汇编代码
  mov eax, [esp+argc]
  and eax, 80000007h  // 10000000000000000000000000000111b
  jns short loc_401010
  dec eax // 一减一加，是对模整除(没有余数)的调整
  or eax, 0FFFFFFF8h // 11111111111111111111111111111000b
  inc eax
loc_401010: // 不为负数走这里
  push eax
```




> 有符号被模数 模 -2以及-2的幂时，经过测试，VS2019 的优化方案和模正数 2 以及 2的幂一致。都有一个 "jns" 的跳转指令，代码定式 "**and jns dec or inc**"。

> 但是 VC++6.0 的优化方案不同（正、负个一种方案），取除数为-2的幂时，生成的汇编代码指令为 "**cdq xor sub and xor sub**"，代码示例：

```
printf("%d\n", argc % -8);
// 对应的汇编代码
mov eax, [esp+argc]
cdq
xor eax, edx
sub eax, edx // 对被模数求绝对值 |eax|
and eax, 7 // 对正8进行取模 |eax| % |-8|
xor eax, edx // 为正数，值不变，为负数，相当于取反加一
sub eax, edx
push eax
// 为负数 eax < 0，-(|eax| % |-8|)
```

无分支求绝对值：

```
int abs(int n)
{
    int m = n >> 31; // 得到符号位
    n = n ^ m; // 异或（取反）
    return n-m; // 减-1 --> 加1
}
```

$C = -2^n$

A % C,  C < 0

|A| %|C|

A >=0  A % C = |A| %|C|

A < 0  A % C = -(|A| %|C|)

```
int abs(int n)
{
  int m = n >> 31;
  n = n ^ m;
  return n - m;
}
```

```
mov eax, n      eax >= 0, edx=0
cdq             eax < 0,  edx=-1
xor eax, edx    eax<0,  eax=noteax
sub eax, edx    eax<0, eax=eax--1
                       =eax+1
eax = abs(n)
                eax<0, not(eax)+1
                       neg(eax)
```