

# 取模

## 无符号

- 2的幂

```
// i % n -> and reg, (n - 1)
// 比如: i % 8
and reg, 7

// n % i -> 直接使用的除法, 不优化
// 比如: 8 % i
```

- 非2的幂

```
// 余 = 被除数 - 商*除数
// 比如: i % 7

// n % i -> 直接使用的除法, 不优化
// 比如: 7 % i
```

## 有符号

- 2的幂

```
// and 和 or 指令的数是对应的
// dec 和 inc 是对0进行特殊处理
// i % 8
    and eax, 80000007h // 1000 0000 0000 0000 0000 0000 0000 0111
    jns label1
    dec eax
    or eax 0fffffff8h // 1111 1111 1111 1111 1111 1111 1111 1000
    inc eax

label1:
    ...

// 扩展: 对eax求绝对值
//      当eax为正, 则edx为0
//       $eax \wedge edx = eax \wedge 0 = eax$ 
//       $eax - edx = eax - 0 = eax$ 
//      当eax为负, 则edx为0xffffffff = -1
//       $eax \wedge edx = eax \wedge -1 = \text{not } eax$ 
//       $eax - edx = eax - (-1) = eax + 1$ , 即取反加1
cdq
xor eax, edx
sub eax, edx

// i % -8 -> |i| % |-8|, 符号跟i走
mov eax, i
```

```
cdq
xor eax, edx
sub eax, edx    // 以上相当于abs(eax)
and eax, 7      // eax % 8
xor eax, edx    // not eax
sub eax, edx    // eax + 1, 以上相当于neg(eax)
```

- 非2的幂

```
// 余 = 被除数 - 商*除数
// 比如: i % 7

// n % i -> 直接使用的除法, 不优化
// 比如: 7 % i
```

## 三目运算

- 无优化, 表达式2、3中存在变量就不优化  
产生分支
- 优化

```
// 核心思想, eax == 0? 0 : -1
neg eax
sbb eax, eax

// eax == 100 ? 0 : -1
sub eax, 100
neg eax
sbb eax, eax

// eax == 100 ? 39 : 93
sub eax, 100
neg eax          // if eax == 100,
sbb eax, eax     // eax = 0 else -1
and eax, 93 - 39 // eax = 0, else 93 - 39
add eax, 39      // eax = 39, else 93

// 利用setxx来优化
// eax > 100 ? 39 : 93
cmp eax, 100
setng cl
dec ecx
and ecx, 39 - 93
add ecx, 93

// 高版本利用条件传送movxx reg, imm
// eax > 100 ? 39 : 93
cmp eax, 100
mov eax, 93
mov ecx, 39
```

```
cmovg eax, ecx
```