2.7.1 算术运算

加、减、乘、除、求余

add 加

sub 减 subtract

mul 乘 multiply

div 除 divide、求余

2.7.2 逻辑运算

```
 &     |      ^      ~      <<     >>
and   or    xor    not    shl    shr

_rotl()    _rotr()
rol          ror
```

设 AL=1011 0110
rol al, 1; 表示把 AL 循环左移 1 位
al = _rotl(al, 1); 表示把 AL 循环左移 1 位
结果 AL=0110 1101

以下代码用<< | >>实现了循环左移运算:

```
unsigned long rol(unsigned long x, int n)
{
   return x << n | x >> (sizeof(x)*8-n);
}
```

shl: shift left

shr: shift right
xor: exclusive or
rol: rotate left 循环左移
ror: rotate right 循环右移

```
mov ah, 9Dh; AH=1001 1101
rol ah, 1   ; AH=0011 1011
mov ah, 9Dh; AH=1001 1101
mov cl, 2
rol ah, cl  ; AH=0111 0110
```

位运算的作用：

① 与运算可以使某些位变 0；

设 a 是一个 8 位数，要使该数的最低位与最高位都变 0，其它位不变：

```
?011 011?
0111 1110  and)
-----------
0011 0110
```

② 或运算可以使某些位变 1；

```
?011 011?
1000 0001 or)
-----------
1011 0111
```

③ 异或运算可以使某些位反转；

```
0011 0111
1000 0001  xor)
-----------
1011 0110
```

运用 rol 指令把 16 位整数转化成 16 进制格式输出：

http://10.71.45.100/bhh/v2h.asm

运用 rol 指令把 32 位整数转化成 16 进制格式输出：

http://10.71.45.100/bhh/v2h32.asm