

mul指令

- 乘法比较特殊， 默认存放在ax当中， 默认是和ax相乘
- 如果发生了溢出， 那么就是将溢出的高位放到dx当中， 低的会放到ax当中

div指令

- 与乘法一致
- 特殊地：
 - 形如 $0x2710/0x64$ ， 结果存放在ax当中（换算成10进制 $10000/100=100$ ）
 - 如果被除数想要大点， 那么形如 $0xf4240/0x2710$ （换算成十进制就是 $1,000,000/10,000$ ）
 - 这种写法就要把被除数的数字前面的放到dx里面
 - 结果商一般存在ax当中， 余数存在dx当中
 - 此外一般为存放地址
 - ax/bl->al
 - DxAx/Bx->Ax

and和or指令（就是c当中的&和|）

- 直接使用and 寄存器， 值 这种格式进行计算
- 结果存放在寄存器当中

shl和shr指令（就是位运算）

- shl左移， shr右移
- 注意： 都是转换成二进制的移动
- 额外的： 有些编译器有循环移动
 - 比如本该舍去的一位会换到新增的那里去

inc和dex指令

- 等价于c语言的++与--， 如果越位就会溢出

nop指令

- 空指令

xchg指令（就是exchange简写）

- 就是交换，没什么好说的

neg指令

- 取反指令，在二进制当中所有的都换以下，0变成1,1变成0

int中断指令（这个和c语言的整型不是一个意思）

- 停止程序例如int 0，与除法当中除以0等价
- 这边本质上是将执行代码层面的cs指向的指针变化到一个0000的位置
- 如果在000:000这个地方写一段代码，可以进行一些操作
- 特殊地，中断不止int 0中断，还有例如：
 - int 21：表示退出程序，后面无论写什么代码都不会再执行了