

数据处理的两个基本问题

sreg 段地址寄存器

reg 寄存器

bx,si,di和bp

- bx si/di组合
- bp si/di组合

指令要处理的数据

1. 保存在CPU
2. 在内存中
3. 在端口中

数据位置的表达

1. 立即数(idata)
2. 寄存器
3. 段地址加偏移地址

数据的长度

- byte 和 word

在处理数据的时候要告知 CPU 要处理的数据有多大可以通过一些方法来告知

1. 通过寄存器来指明 如 **ax**, 代表对word操作而 **al**, 代表对byte 操作

2. 无寄存器 则用 **X ptr** 来表示 X 为byte 或者 word 如 : `mov word ptr ds:[0],1`

3. 用 **push or pop** 就不用声明 因为 栈就是对字进行操作

div 指令

- 除数 有8位和16位 在一个reg或内存单元中
- 被除数 默认 放在 **ax (16位)** 或者 **dx (高16位)** 和 **ax (低16位)** 中
- 结果 **al(商) ah(余数)** 或者 **ax(商) dx(余数)**

伪指令 dd (占两个字)

- 相当于 两个 **dw**
- 四个 **db**

dup

用来重复数据

- `db 3 dup (0) ==> db 0,0,0`