

一些汇编的小坑

张艺瀚

1 基本语法

- `ADDR[BX][SI]`, `ADDR[SI][BX]`, `0100H[BX][SI]`的寻址方式都是正确的。
- 一般来说, 立即寻址只能用于源操作数寻址。
- 除源操作数为立即寻址的指令外, 两个操作数的指令, 其中必须有一个操作数的寻址方式为寄存器直接寻址。
这样写是错误的: `MOV BYTE PTR [DI], [SI]`。
- 在算术表达式中, 除`+`, `-`, `*`, `/`外, 只能使用`MOD`, `SHL`, `SHR`, `AND`, `OR`, `XOR`, `NOT`, 它们在表达式中是作为逻辑算符存在的, 表达式实在将源程序翻译成目标程序时求值的, 而当它们出现在指令中时, 要在执行目标程序时求值的。
- `SIZE`和`LENGTH`只适用于`DUP`分配的内存单元。
- `MOV AX, [0100H]`将被反汇编成`MOV AX, 0100H`。
- `LEA`的目的操作数可以为任意16位通用寄存器或指针和变址寄存器, 用`LEA`指令获取偏移量和用`OFFSET`算符获取偏移量的区别在于, `OFFSET`只能跟标号。
比如我们可以写: `LEA AX, [BX + SI + 0100H]`,
但不能写: `MOV AX, OFFSET [BX + SI + 0100H]`。
- `DS`, `ES`, `SS`中均可以存放指令, 存放的是指令的编码, 但不会被执行到, 所以课本上说指令代码只能存在`CS`中不能算错。如果想看指令编码可以反汇编, 不要用这种反人类方式。
- `DEBUG`
 - `U`命令中可以给出两个地址, 它们必须是偏移量。

- T命令格式: T [=地址] [,计数] (可以指定步长)。
- P命令不进入CALL, INT和循环。
- R后给出寄存器名称可以修改其值。
- D默认显示128字节内存, 且可以给出的两个地址必须是偏移量。
- A命令下可以直接在指令中用偏移地址的值寻址, 但将这些命令写到.ASM文件中都是错的。
如MOV AH, [0100H]被翻译成MOV AH, 0100, JMP 0100H会报错。

2 顺序结构

- MOV AX, LABEL的写法是错误的。
- ADD, ADC, SUB, SBB, CMP, AND, OR, XOR等指令的操作数中不得出现段寄存器。事实上, 除传送指令, 其他均不能用段寄存器。
- MOV [BP + OFFSET DATA], AH这种写法默认SS寻址。
MOV DATA[BP], AH这种写法默认DS寻址。将源程序翻译为目标程序时, 翻译程序在DATA[BP]前面自动加上DS:。
- 对标志位的特殊影响
 - NEG操作前操作数非0, 则操作后CF置位, 否则复位。8位时的80H, 16位时的8000H, NEG后溢出, OF被置位。
 - 执行MUL后高位为0或执行IMUL高位是低位符号位的扩展时 (此时高位可丢弃), CF, OF被复位。
另外, MUL, IMUL, DIV, IDIV源操作数不能为立即数。
 - DIV, IDIV不产生有效的标志位影响, 各位不定。若溢出则产生0中断, 结果不定。
 - CBW, CWD不影响标志位。
 - NOT不影响标志位。
AND, OR, XOR, TEST操作后, CF, OF被复位, 除AF不定外, 其他位正常影响。
 - 移位操作后, 最高位若改变则OF被置位 (只适用于移位1位的情况)。

3 分支结构

- 以下写法都是正确的:
 - JMP LABEL

- JMP AX
- JMP MEM

JMP [BX]的写法是正确的，默认段间转移，但写成：JMP WORD PTR [BX]（段内转移）或JMP DWORD PTR [BX]（段间转移）更好。

但JMP 0100H是错误的。

对于JMP DWORD PTR [BX]，BX所值内存的低位的一个字存放偏移量，高位的一个字存放偏移量，届时前者被送入IP，后者被送入CS。

条件转移指令后面只能跟标号。像JMP那样的JC AX, JC MEM, JC [BX]的写法统统错误。

另外，如果要实现段间转移，LABEL要在其他段中用PUBLIC说明，在本段中用EXTRN说明。

4 循环结构

- 数据传最长64K。
- MOVS, MOVSB, MOVSW, CMPS, CMPSB, CMPSW的两个操作数可均为存储器操作数。
串操作指令操作数的写法：

- MOVS ES: DATA2, DS: DATA1
- CMPS DS: DATA1, ES: DATA2
- SCAS ES: DATA2
- LODS DS: DATA1
- STOS ES: DATA2

上述指令格式中，DS:可省略，ES:不可省略。因为默认DS段寻址。

- 串操作指令的执行情况：
 - CMPS, CMPSB, CMPSW: DS: [SI] - ES: [DI]
 - SCAS, SCASB, SCASW: AL / AX - ES: [DI]
 - LODS, LODSB, LODSW: AL / AX ← DS: [SI]
 - STOS, STOSB, STOSW: ES: [DI] ← AL / AX

MOVS, MOVSB, MOVSW, CMPS, CMPSB, CMPSW, LODS, LODSB, LODSW, STOS, STOSB, STOSW不影响标志位。

使用串操作指令时可能需要预设的内容：

- DF

- SI, DI
- CX
- AL / AX
- LOOP, LOOPE, LOONE指令与循环入口的距离为-126 129字节。其后只能跟标号，像JMP那样的LOOP AX, LOOP MEM, LOOP [BX]的写法统统错误。这些指令不影响标志位。所以先将CX减1再无条件转移到循环入口和直接LOOP是有区别的。
- REP: 先判断CX状态，再执行，再将CX减1。
LOOP: 先执行，再将CX减1，再判断CX状态。
所以若将CX置0，REP修饰的指令执行0次，LOOP执行65536次。

5 子程序

- 用PROC定义的子过程中不能使用HLT等引起停机的指令。

- 以下写法都是正确的:

- CALL PN
- CALL AX
- CALL MEM

只写CALL PN，不标明NEAR或FAR，默认近调用。

CALL [BX]的写法是正确的，默认调用近过程，但这是做死的，强烈建议写成: CALL WORD PTR [BX]（调用近过程）或CALL DWORD PTR [BX]（调用远过程）。

MEM若为WORDL类型，则为近调用，若为DWORD类型，则为远调用。

- 产生远调用时:
 - CS压栈
 - 低字所存偏移量送CS
 - IP压栈
 - 高字所存段地址送IP

产生近调用时，没有前两步。

- 远调用不能用段寄存器指定入口。

