BIN/COM格式与加载

一、纯码格式

BIN/COM格式的可执行程序都是纯码，即内容由源程序中的指令机器码和数据组成。

以下面的汇编源码为例，

mov ax,0xB800

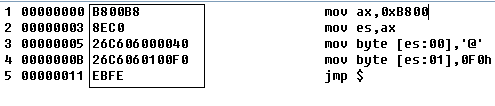
mov es,ax

mov [es:0],’@’

mov [es:1],7

jmp $

观察汇编产生的列表文件，知道程序产生的纯码有19个字节，见下图框内：



所以，nasm汇编产生的bin/COM程序，文件内就只有这框内的19个字节内容，没有任何其他内容。

二、ORG伪指令的作用

ORG伪指令指示汇编器对段内符号地址的安排。

在汇编源码中，可以定义多个独立的段。前面这个程序，只有一个段，即代码段，没有定义其他段。

汇编器在安排符号地址偏移量时，默认是偏移段首的值。

例如，以下面的汇编源码，没有使用ORG伪指令，有两个符号地址：x,y

mov ax,0xB800

mov es,ax

mov [es:0],’@’

mov [es:1],7

jmp $

x dw 10

y dw 20

那么，汇编器对两个符号地址x和y安排的地址就是它们偏移本段首的偏移量：即x为19，y为21。

如果修改一下汇编源码，程序使用ORG伪指令，汇编器对两个符号地址x和y安排的地址就有变化：

ORG 100

mov ax,0xB800

mov es,ax

mov [es:0],’@’

mov [es:1],7

jmp $

x dw 10

y dw 20

那么，汇编器对两个符号地址x和y安排的地址就是它们偏移本段首的偏移量+100：即x为119，y为121。如果程序中有指令引用x和y，那么，相应位置的指令汇编结果也就有变化。

产生COM格式的汇编源程序，约定要用ORG 100h伪指令。例如，下面程序中的符号地址(如x和y)安排的地址就是它们偏移本段首的偏移量+100h(即十进制为256)：x和y分别为256+19和256+21。

ORG 100h

mov ax,0xB800

mov es,ax

mov [es:0],’@’

mov [es:1],7

jmp $

x dw 10

y dw 20

三、bin/COM格式程序

如果程序中，没有定义符号地址(又称标号)，ORG就不起作用。实验中，一些同学的bin/COM格式程序都侥幸能DOS里面运行。

Bin格式的程序，监控程序加载和跳转相当简单，比如加载到内存8000h开始的位置，跳转指令可以写成：jmp 0x800:0或jmp 0:8000h，应该跳转没有问题。

COM格式大小不能超过64k而且运行条件是cs=ds=ss=es，即所谓4段重叠。所以，汇编源码的开头几条指令，通常如下程序所示：

1. org 100h
2. start: mov ax, cs
3. mov ds, ax ; DS = CS
4. mov es, ax ; ES = CS
5. mov ss, ax ; SS = CS

COM格式的程序，监控程序加载和跳转就要主要匹配，否则运行会不正常，最常见的是显示乱码。有一个COM格式的程序，比如分配内存段起始地址为8000h，那么程序要加载到内存8000h+100h开始的位置，跳转指令可以写成：jmp 0x800:100。这样，跳转时，cs=8000h且ip=100h，再执行COM程序开头几条指令，保证cs=ds=ss=es=8000h，那么程序中的引用符号变量的指令就能正确执行。注意：实际上段寄存器不保存后4位，所以cs、ds、ss、es内容为800h。