Lab7

Hyunki Lee

Exercise1

mov ax,[arrayW+6] ax: 00190001 ; word is 2byte therefore the movement of offset is 2 (0,2,4 ~~). There are 3 initializations, thus after +4, the first value of arrayD which is 1 is stored in ax.

mov eax,[arrayD-4] eax: 30002000 ; ‘-4’ means it goes to the back direction and move 4 byte. The offset arrives at 2000h (arrayW). The destination is eax which is 4byte, therefore, 2000h and 3000h is stored in eax.

mov eax,[arrayD+12] eax: 4 ; It is dword. Thus it move 4bytes. Therefore, +12 indicates the 4th value of arrayD which is 4

mov eax,[arrayD+16] ax: 120B0890 ; This is 4bytes. The offset move +4. The +16 indicates arrayB 90h. and arrayB is byte size. Therefore, 4byte which is 4 values from arrayB is stored in eax.(90h,08h,11,12h)

mov ax,[arrayW+22] ax: 120B0890 ; The 22th value from arrayW is 90h from arrayB. Therefore, 90h,08h is stored in ax.

mov ecx,[arrayD+2] ecx: 00020000 ; This is 4bytes, it will be stored 4bytes size from offset 2 position. Therefore, from offset2 to offset5

|  |  |
| --- | --- |
| stored | Offset |
| 01 | 0 |
| 00 | 1 |
| 00 | 2 |
| 00 | 3 |
| 20 | 4 |
| 00 | 5 |

Exercise3

mov bx,0A69h

movzx eax,bx ;eax: 00000A69

movzx eax,myByte1 ;eax: 0000009B

mov bx,0A69Bh

movsx eax,bx ;eax: FFFFA69B

movsx eax,ebx ;error

movsx is for signed number, and it uses for the two different operands’ size.

In this case, eax and ebx is the same size, therefore, the movsx does not work.

Exercise 4-1

mov ax,7ff0h ;SF(PL)=0 ZF=1 CF=0 OF=0 ;

add ax,10h ;SF(PL)=1 ZF=0 CF=0 OF=1 ;

add ah,1 ;SF(PL)=1 ZF=0 CF=0 OF=0 ;

add ax,2 ;SF(PL)=1 ZF=0 CF=0 OF=0 ;

Yes. By using neg operation, the signed number can be changed as unsigned number or unsigned number can be signed number.

Exercise 4-2

1. CF=0 SF=1
2. CF=0 SF=1
3. SF=1 OF=1
4. SF=1 OF=1

Exercise 4-3

mov al,-128 ; AL=0019FF80

neg al ; AL=0019FF80 OF=1

mov bl,0

neg bl ; BL=003DD000 SF= 0 ZF=1 OF=0