## Section #03

# (Writing assembly using Debug)

### 1) $A \rightarrow Assemble$ .

A Assemble Used to enter assembly language.

```
C:\>debug

-a100

073F:0100 mov ax,1

073F:0103 mov bx,2

073F:0106 mov cx,3

073F:0109 add ax,bx

073F:010B add ax,cx

073F:010D int 3

073F:010E
```

### 2) $U \rightarrow Unassemble$ .

- Can be written in two formats [U StartA EndA, U StartA L#bytes].
- You cannot assemble lower than 100 because the beginning 256 bytes are reserved by DOS.

```
-u 100 10d
073F:0100 B80100
                         MOV
                                 AX,0001
073F:0103 BB0200
                         MOV
                                 BX,0002
                         MOV
073F:0106 B90300
                                 CX,0003
073F:0109 01D8
                         ADD
                                 AX, BX
073F:010B 01C8
                         ADD
                                 AX,CX
073F:010D CC
                         INT
```

# 3) $G \rightarrow Go$

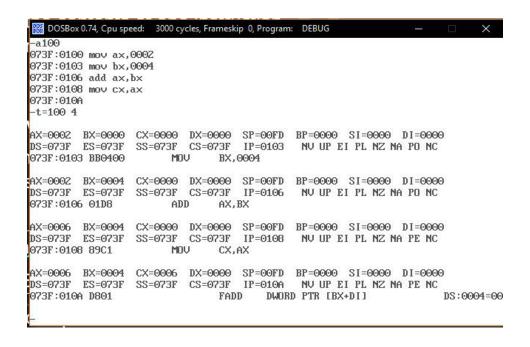
- Execute instructions between two addresses.
- If no address begins execute instruction at CS:IP until a breakpoint is reached.
- INT 3 terminate execution.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
D:N>debug
-a100
973F:0100 mov ax,2
973F:0103 mov bx,3
973F:0106 add ax,bx
973F:0108 int 3
973F:0109
4X=0005 BX=0003 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000 DS=073F ES=073F SS=073F CS=073F IP=0108 NV UP EI PL NZ NA PE NC
973F:0108 CC
-g=100 103
AX=0002 BX=0003 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000 DS=073F ES=073F SS=073F CS=073F IP=0103 NV UP EI PL NZ NA PE NC
                                                       NU UP EI PL NZ NA PE NC
973F:0103 BB0300
                            MOV
                                      BX,0003
-g=100
AX=0005 BX=0003 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F
                                           IP=0108
                                                       NU UP EI PL NZ NA PE NC
973F:0108 CC
                             INT
```

## **4) T**→ **Trace**

- Trace the contents of one instruction.
- Can write in form [T begin address #of Instruction].

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
C:\>debug
-a100
073F:0100 mov ax,0002
073F:0103 mo∨ b×,0004
073F:0106 add ax,bx
073F:0108 nop ; No operation
073F:0109
-t
AX=0002 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103 NV UP EI PL NZ NA PO NC
                      MOV
073F:0103 BB0400
                              BX.0004
AX=0002 BX=0004 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS-073F ES-073F SS-073F CS-073F IP-0106 NV UP EI PL NZ NA PO NC
                      ADD
073F:0106 01D8
                              AX.BX
AX-0006 BX-0004 CX-0000 DX-0000 SP-00FD BP-0000 SI-0000 DI-0000
DS-073F ES-073F SS-073F CS-073F IP-0108 NV UP EI PL NZ NA PE NC
073F:0108 90
```



#### True or False:



## 5) $W \rightarrow Write$

# 6) $N \rightarrow Name$

- Steps used to save an assembly program:
- Obtain length using "h"
- Name your program using "n"
- Put length on "CX"
- Write your program on disk using "w"

```
C:\8086>debug

-a100

073F:0100 mov ah,02

073F:0102 mov dl,01

073F:0104 int 21

073F:0106 int 20

073F:0108

-h108 100

0208 0008

-n smile.com

-r cx

CX 0000

:0008

-w

Writing 00008 bytes
```

# 7) L Load

- First give the name of the file you want to load.
- Load file using "L"

```
C:\8086>debug
-n smile.com
-1
-u 100 108
075A:0100 B402
                          MOV
                                  AH,02
075A:0102 B201
                                  DL,01
                          MOV
075A:0104 CD21
                          INT
                                  21
075A:0106 CD20
                                  20
                          INT
075A:0108 0000
                          ADD
                                  [BX+SI],AL
```

# Q→ Quit.

- Exit from debug program.