

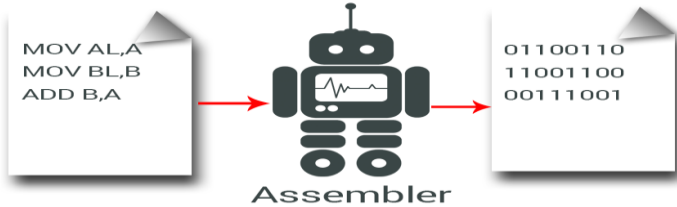


Mansoura University
Faculty of Computers and Information
Department of Computer Science
First Semester: 2020-2021



[CS214P] Assembly Language
Grade: Third Year (Computer Science)

Sara El-Metwally, Ph.D.
Faculty of Computers and Information,
Mansoura University,
Egypt.



Computer Science Department
Faculty of Computers and Information
Mansoura University

Assembly Language

"Examining Computer Memory and Executing Instructions"

Sara El-Metwally, Ph.D.
Faculty of Computers and Information,
Mansoura University, Egypt.

Email: sarah_almetwally4@mans.edu.eg
sara.elmetwally.2007@gmail.com

Debug Program

- **DEBUG** program allows you to view memory, to enter programs in memory, and to trace their execution.
- **DEBUG** program is used for testing and debugging executable programs.
- **DEBUG** displays all program code and data in hexadecimal format.
- **DEBUG** has a single-step mode, which allows you to execute a program one instruction at a time.

Debug Commands

- **A:** Assemble symbolic instructions into machine code.
- **D:** Display the contents of an area of memory in hex format.
- **E:** Enter data into memory, beginning at a specific location.
- **G:** Run the executable program in memory (G means “Go”).
- **H:** Perform hexadecimal arithmetic.

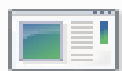
Debug Commands

- **N:** Name a program.
- **P:** Proceed, or execute a set of related instructions.
- **R:** Display the contents of one or more registers in hex format.
- **T:** Trace the execution of one instruction.
- **U:** Unassemble (or disassemble) machine code into symbolic code.

Rules of Debug Commands

- **DEBUG:** lowercase = uppercase letters.
- **DEBUG:** assumes that all numbers in hexadecimal format.
- **DEBUG:** spaces in commands are used only to separate parameters.
- **DEBUG :** segments and offsets are specified with a colon, in the form segment: offset

Your Working environment



debug

Type: Application

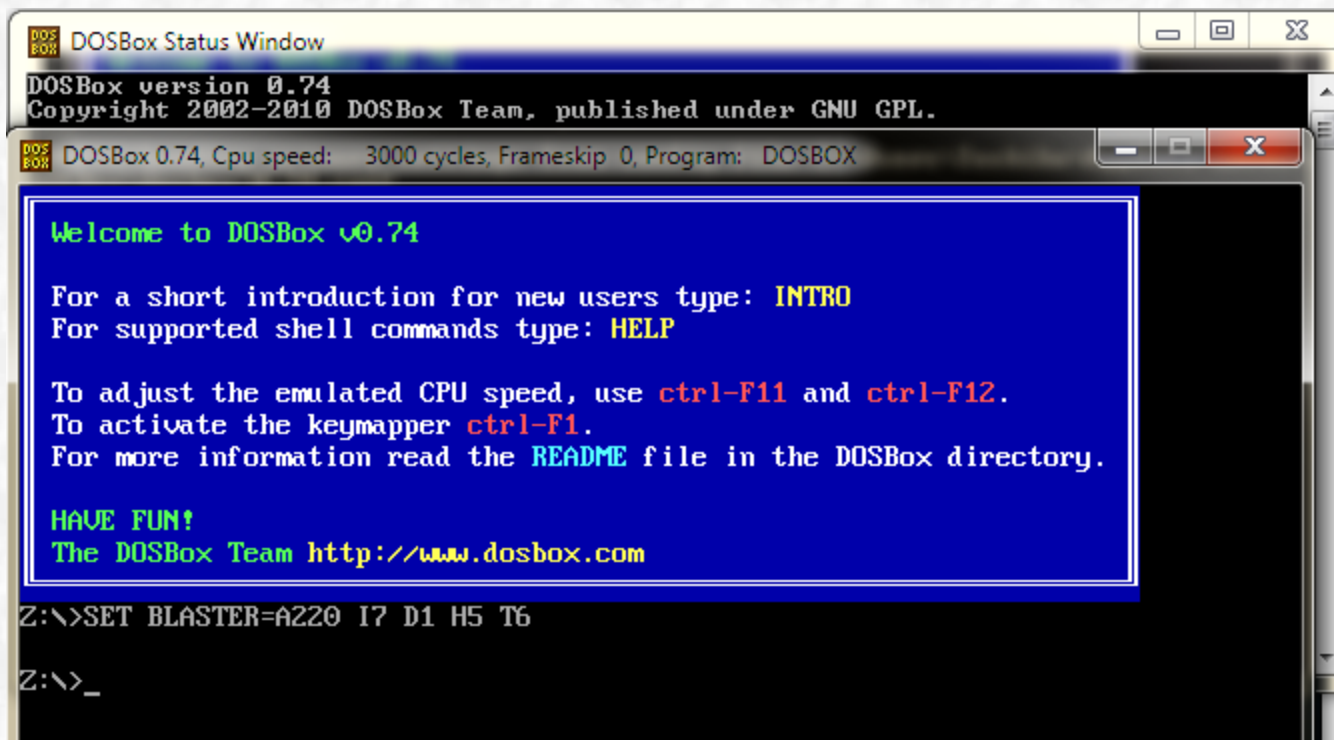


DOSBox0.74-win32-installer

Type: Application

Your Working environment

- Install DOSBox
- Open DOSBox



Your Working environment

- **Type:**

```
mount <space> c <space> c:\
```

```
mount <space> c <space> c:\Documents\
```



could be any chars

Location of debug.exe

- **Then, change the path to the named directory i.e. c .**
- **Type: debug.exe**

Your Working environment

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

Here are some commands to get you started:
Before you can use the files located on your own filesystem,
You have to mount the directory containing the files.

`mount c c:\dosprogs\` will create a C drive with `c:\dosprogs` as contents.
`c:\dosprogs\` is an example. Replace it with your own games directory.

When the mount has successfully completed you can type `c:` to go to your freshly mounted C-drive. Typing `dir` there will show its contents. `cd` will allow you to enter a directory (recognised by the `[]` in a directory listing).
You can run programs/files which end with `.exe`, `.bat` and `.com`.

`Z:\>mount y D:\College-Courses\Assembly\debug\`
Drive Y is mounted as local directory `D:\College-Courses\Assembly\debug\`

`Z:\>y:\`

`Y:\>debug.exe`
`-a100`
`073F:0100 _`

Annotations:

- `name` points to `y` in `mount y`
- `location of my debug.exe` points to `D:\College-Courses\Assembly\debug\`
- `change path to the new name` points to `y:\`
- `execute debug.exe` points to `debug.exe`

Debug Display Command

```
C:\>DEBUG.EXE
-D DS:200
073F:0200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
073F:0210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
073F:0220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
073F:0230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
073F:0240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
073F:0250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
073F:0260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
073F:0270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....
```

The address of left
most displayed byte
segment : offset

Hex representation
of the displayed
area.

ASCII representation
of the displayed area.

Debug Display Command (BIOS data area, 40[0]H)

Address of first
serial port (?)

Address of second
serial port (?)

Address of one
parallel port (?)

-D 40:00																	
0040:0000	F8	03	F8	02	00	00	00	00	F8	03	00	00	00	00	00	00x.....
0040:0010	2B	D4	00	80	02	00	00	40	00	00	2E	00	2E	00	44	20	&.....@.....D
0040:0020	20	39	34	05	30	0B	3A	27	30	0B	30	0B	0D	1C	71	10	94.0.: '0.0...q.
0040:0030	0D	1C	E0	48	E0	48	0D	1C	E0	48	E0	48	0D	1C	00	00	...H.H...H.H....
0040:0040	00	00	00	00	00	00	00	00	00	03	50	00	00	10	00	00P.....
0040:0050	00	08	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0040:0060	07	06	00	D4	03	29	30	00	00	00	00	48	C9	12	00	00)0.....H...
0040:0070	00	00	00	00	00	02	00	00	01	01	01	00	01	01	01	01

4 words : addresses of
serial ports COM1
through COM4.

The first 16 bytes: the addresses of the
serial and parallel ports.

4 words : addresses of
parallel ports LPT1
through LPT4.

Debug Display Command (BIOS data area, 40[0]H)

Equipment status word **D426**
(indication of installed devices)

-D 40:10

```
0040:0010 26 D4 00 80 02 00 00 40-00 00 22 00 22 00 30 0B &.....e..".".0.
0040:0020 0D 1C 34 05 30 0B 3A 27-30 0B 30 0B 0D 1C 44 20 ..4.0.: '0.0...D
0040:0030 20 39 31 02 08 0E 34 05-30 0B 3A 27 31 02 00 00 91...4.0.: '1...
0040:0040 00 00 00 00 00 00 00 00-00 03 50 00 00 10 00 00 .....P.....
0040:0050 00 10 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0040:0060 07 06 00 D4 03 29 30 00-00 00 00 00 6F 1A 14 00 .....0.....o...
0040:0070 00 00 00 00 00 02 00 00-01 01 01 00 01 01 01 01 .....
0040:0080 1E 00 3E 00 18 10 00 60-09 51 0B 00 00 00 00 00 ..>....` .Q.....
```

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	0	0	0	0	1	0	0	1	1	0

Debug Display Command (BIOS data area, 40[0]H)

D426

Numeric coprocessor is present =1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	0	0	0	0	1	0	0	1	1	0

Number of parallel printer
ports attached

Number of serial ports
attached

Diskette drive is
present =1

Initial video mode =10
01=40x25 color
10=80x25 color
11=80x25 monochrome

Number of diskette
devices attached
00=1, 01=2, 10=3, 11=4

Debug Display Command

(ROM BIOS, copyright, FE00:0)

-D FE00:0000

FE00:0000	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 49 42IB
FE00:0010	4D 20 43 4F 4D 50 41 54-49 42 4C 45 20 34 38 36	M COMPATIBLE 486
FE00:0020	20 42 49 4F 53 20 43 4F-50 59 52 49 47 48 54 20	BIOS COPYRIGHT
FE00:0030	54 68 65 20 44 4F 53 42-6F 78 20 54 65 61 6D 2E	The DOSBox Team.
FE00:0040	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
FE00:0050	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
FE00:0060	00 44 4F 53 42 6F 78 20-46 61 6B 65 42 49 4F 53	.DOSBox FakeBIOS
FE00:0070	20 76 31 2E 30 00 00 00-00 00 00 00 00 00 00 00	v1.0.....

Debug Display Command

(ROM BIOS Date, FFFF:5)

D FFFF:0005

```
FFFF:0000      30 31 2F-30 31 2F 39 32 00 FC 55      01/01/92..U
FFFF:0010  60 10 00 F0 08 00 70 00-08 00 70 00 08 00 70 00  ..p..p..p.
FFFF:0020  08 00 70 00 60 10 00 F0-60 10 00 F0 60 10 00 F0  ..p. ....
FFFF:0030  A5 FE 00 F0 87 E9 00 F0-55 FF 00 F0 60 10 00 F0  .....U....
FFFF:0040  60 10 00 F0 60 10 00 F0-80 10 00 F0 60 10 00 F0  .....
FFFF:0050  00 13 00 F0 00 11 00 F0-20 11 00 F0 40 11 00 F0  .....@...
FFFF:0060  A0 11 00 F0 C0 11 00 F0-E0 11 00 F0 20 12 00 F0  .....
FFFF:0070  C0 12 00 F0 C0 12 00 F0-40 12 00 F0 60 10 00 F0  .....@...
FFFF:0080  60 12 00 F0 A4                                     ....
```

Machine Language Example

(Using Immediate Data)

B82301

052500

8BD8

03D8

8BCB

2BC8

2BC0

EBEE

Machine Instruction

MOV AX, 0123

ADD AX, 0025

MOV BX, AX

ADD BX, AX

MOV CX, BX

SUB CX, AX

SUB AX, AX

JMP 100

Symbolic Code

Machine Language Example

(Using Immediate Data)

First 6 bytes of machine codes
starting from 100 ending with 105

B82301

052500

8BD8

03D8

8BCB

2BC8

2BC0

EBEE

Machine Instruction

E	CS:100	B8	23	01	05	25	00
---	--------	----	----	----	----	----	----

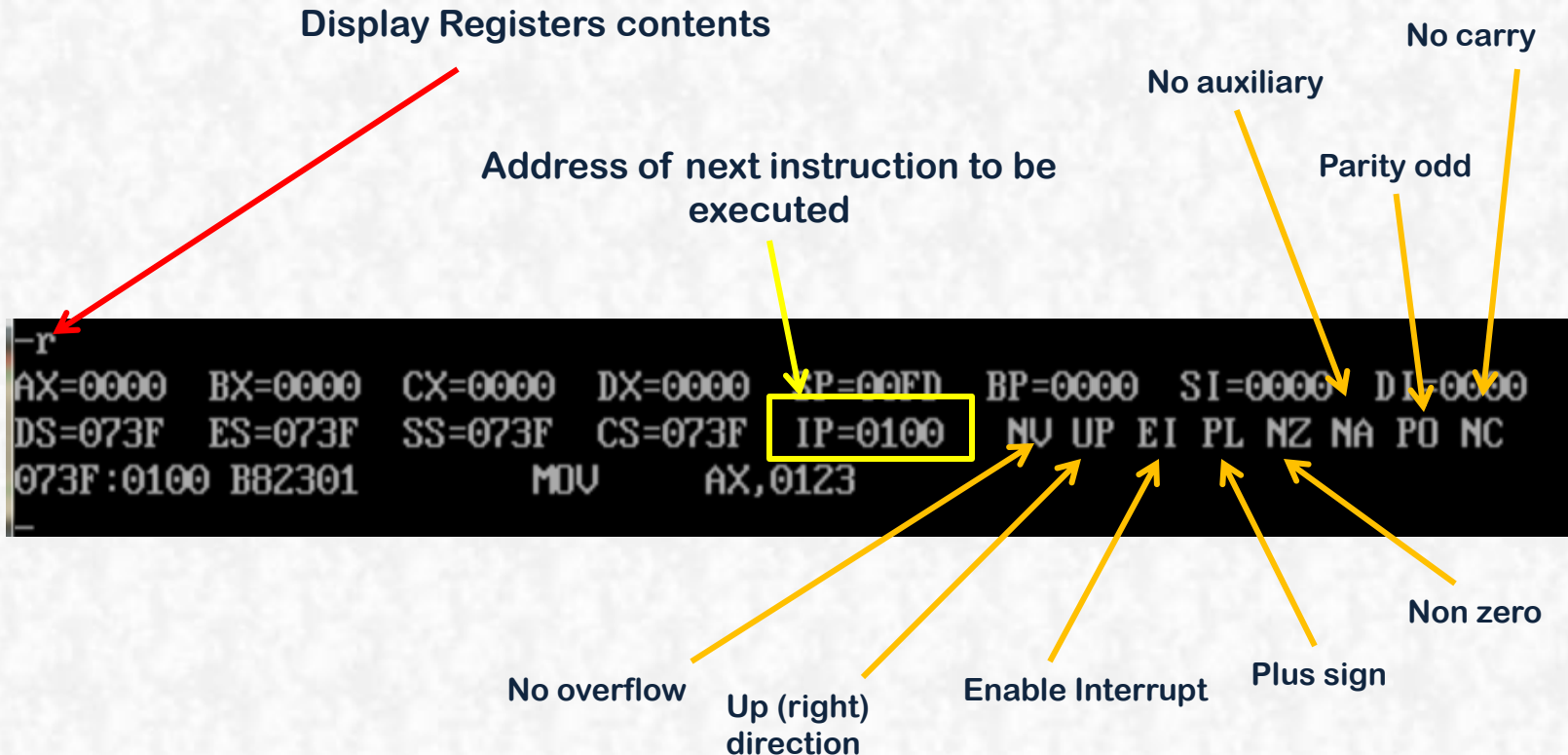
Starting memory address CS:100

Enter data into memory beginning at
specific location

E	CS:100	B8	23	01	05	25	00
E	CS:106	8B	D8	03	D8	8B	CB
E	CS:10C	2B	C8	2B	C0	EB	EE
E							

Machine Language Example

(Using Immediate Data)



Machine Language Example

(Using Immediate Data)

Trace the execution of one instruction

073F:0100 B82301

MOV AX,0123

-T

AX=0123 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

073F:0103 052500

ADD AX,0025

Machine Language Example

(Using Immediate Data)

```
073F:0100 B82301      MOV     AX,0123
-T
AX=0123 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103  NU UP EI PL NZ NA PO NC
073F:0103 052500      ADD     AX,0025
-T
AX=0148 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0106  NU UP EI PL NZ NA PE NC
073F:0106 8BD8        MOV     BX,AX
```

Machine Language Example

(Using Immediate Data)

```
073F:0106 8BD8      MOV     BX,AX
-T

AX=0148 BX=0148 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
073F:0108 03D8      ADD     BX,AX
-T

AX=0148 BX=0290 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=010A  NU UP EI PL NZ AC PE NC
073F:010A 8BCB      MOV     CX,BX
-T

AX=0148 BX=0290 CX=0290 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=010C  NU UP EI PL NZ AC PE NC
073F:010C 2BC8      SUB     CX,AX
```

Reset IP value (R IP)

```
Y:\>DEBUG.EXE  
-R IP  
IP 0100  
:_
```

```
Y:\>DEBUG.EXE  
-R IP  
IP 0100  
:0200  
-R  
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000  
DS=073F ES=073F SS=073F CS=073F IP=0200  NV UP EI PL NZ NA PO NC
```


Assembly Language Program

(A: assemble command)

```
-A 100  
073F:0100
```

A command tells DEBUG to begin accepting symbolic assembly instructions and to convert them into machine code.

```
-A 200  
073F:0200 MOV CL,42  
073F:0202 MOV DL,2A  
073F:0204 ADD CL,DL  
073F:0206 JMP 100  
073F:0209
```

```
-G=200 206
```

```
AX=0000 BX=0000 CX=006C DX=002A SP=00FD BP=0000 SI=0000 DI=0000  
DS=073F ES=073F SS=073F CS=073F IP=0206  NV UP EI PL NZ NA PE NC  
073F:0206 E9F7FE          JMP     0100  
-
```

Assembly Language Program

(U: unassemble command)

```
-A 100
073F:0100 MOV AL,[0016]
073F:0103
-U 100,102
073F:0100 A01600      MOV     AL,[0016]
-
```