

# Lab Task 2

Name : M.Shafeen

Roll No : 22P-9278

## Step : 1

First we RUN the “afd c01-01.com” command to run the file named c01-01 in the DEBUGGER

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0000 SI 0000 CS 19F5 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0012 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF 0 DF 0 IF 1 SF 0 ZF 0 AF 0 PF 0 CF 0
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00

CMD >

0100 B80500 MOV AX,0005
0103 BB0A00 MOV BX,000A
0106 01D8 ADD AX,BX
0108 BB0F00 MOV BX,000F
010B 01D8 ADD AX,BX
010D B804C MOV AX,4C00
0110 CD21 INT 21
0112 0000 ADD [BX+SI],AL

1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00

2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = f.Ω≡ i |..+...
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF .....ft. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 .....δ.L.
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 6.....J. ....
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri
```

## Explanation Of the DEBUGGER:

### M1 :

It displays the MEMORY of the CODE in our computer ( DATA )

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0000	SI 0000	CS 19F5	IP 0100	Stack +0 0000	Flags 7202
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0012	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 0 0 0 0

CMD >

1

	8	9	A	B	C	D	E	F
DS:0008	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01
DS:0018	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
DS:0028	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19
DS:0038	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00
DS:0048	00	00	00	00	00	00	00	00
DS:0050	CD	21	CB	00	00	00	00	00

2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00	= f.Ω≡ i  ..+...
DS:0010	18	01	10	01	18	01	92	01	01	01	00	02	FF	FF	FF	FF	.....ff. ....
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11	δ. L.
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00	ó.....J. ....
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

1 Step

2ProcStep

3Retrieve

4Help ON

5BRK Menu

6

7 up

8 dn

9 le

10 ri

As you can see in the DS:0008 moved from address DS:0000

M2 :

M2 represents the memory of our CODE as a WHOLE ( CODE )

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0000	SI 0000	CS 19F5	IP 0100	Stack +0 0000	Flags 7202
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0012	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 0 0 0 0

CMD >

1

	8	9	A	B	C	D	E	F
DS:0008	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01
DS:0018	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
DS:0028	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19
DS:0038	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00
DS:0048	00	00	00	00	00	00	00	00
DS:0050	CD	21	CB	00	00	00	00	00

2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
DS:0010	18	01	10	01	18	01	92	01	01	01	00	02	FF	FF	FF	FF	.....ff. ....
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11	δ. L.
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00	ó.....J. ....
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
DS:0050	CD	21	CB	00	00	00	00	00	00	00	00	00	20	20	20	20	=!¶.....

1 Step

2ProcStep

3Retrieve

4Help ON

5BRK Menu

6

7 up

8 dn

9 le

10 ri

## Registers :

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0000 SI 0000 CS 19F5 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0 0

CMD >

0100 B80500 MOV AX,0005
0103 B80A00 MOV BX,000A
0106 01D8 ADD AX,BX
0108 B80F00 MOV BX,000F
010B 01D8 ADD AX,BX
010D B8004C MOV AX,4C00
0110 CD21 INT 21
0112 0000 ADD [BX+SI],AL

1 8 9 A B C D E F
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0050 CD 21 CB 00 00 00 00 00

2 F 0 1 2 3 4 5 6 7 8 9 A B C D E
DS:FFFF 00 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00 . = f.Ω i | . + . .
DS:000F 00 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF . . . . . ff . . . . .
DS:001F FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6 . . . . . J . . . . . δ.μ
DS:002F 11 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 . 6 . . . . . J . . . . .
DS:003F 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00 . . . . . . . . . .

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

```

## Code Section / Instruction Section / Instructions in MEMORY :

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0000	SI 0000	CS 19F5	IP 0100	Stack +0 0000	Flags 7202
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 0 0 0 0

CMD >

0100 B80500	MOV	AX,0005
0103 B80A00	MOV	BX,000A
0106 01D8	ADD	AX,BX
0108 B80F00	MOV	BX,000F
010B 01D8	ADD	AX,BX
010D B804C	MOV	AX,4C00
0110 CD21	INT	21
0112 0000	ADD	[BX+SI],AL

1	8	9	A	B	C	D	E	F
DS:0008	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01
DS:0018	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
DS:0028	FF	FF	FF	FF	EB	19	E6	11
DS:0030	A2	01	14	00	18	00	F5	19
DS:0038	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00
DS:0048	00	00	00	00	00	00	00	00
DS:0050	CD	21	CB	00	00	00	00	00

2	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
DS:FFFF	00	CD	20	FF	9F	00	EA	FF	FF	AD	DE	1B	05	C5	06	00
DS:000F	00	18	01	10	01	18	01	92	01	01	01	01	00	02	FF	FF
DS:001F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	E6
DS:002F	11	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00
DS:003F	00	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

## Norse Methodology Symbols :

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0000	SI 0000	CS 19F5	IP 0100	Stack +0 0000	Flags 7202
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 0 0 0 0

CMD >

0100 B80500	MOV	AX,0005
0103 B80A00	MOV	BX,000A
0106 01D8	ADD	AX,BX
0108 B80F00	MOV	BX,000F
010B 01D8	ADD	AX,BX
010D B804C	MOV	AX,4C00
0110 CD21	INT	21
0112 0000	ADD	[BX+SI],AL

1	8	9	A	B	C	D	E	F
DS:0008	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01
DS:0018	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
DS:0028	FF	FF	FF	FF	EB	19	E6	11
DS:0030	A2	01	14	00	18	00	F5	19
DS:0038	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00
DS:0048	00	00	00	00	00	00	00	00
DS:0050	CD	21	CB	00	00	00	00	00

2	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
DS:FFFF	00	CD	20	FF	9F	00	EA	FF	FF	AD	DE	1B	05	C5	06	00
DS:000F	00	18	01	10	01	18	01	92	01	01	01	01	00	02	FF	FF
DS:001F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	E6
DS:002F	11	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00
DS:003F	00	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

## Instruction Pointer ( IP ) :

The instruction pointer tells us what commands are to be run next from MEMORY...

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 000F  SI 0000  CS 19F5  IP 0108  Stack +0 0000  Flags 7204
BX 000A  DI 0000  DS 19F5          +2 20CD
CX 0000  BP 0000  ES 19F5  HS 19F5  +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000  SP FFFE  SS 19F5  FS 19F5  +6 EA00  0 0 1 0 0 0 1 0

CMD >

0106 01D8      ADD     AX,BX
0108 B80F00     MOV     BX,000F
010B 01D8      ADD     AX,BX
010D B8004C     MOV     AX,4C00
0110 CD21      INT     21
0112 0000      ADD     [BX+SI],AL
0114 0000      ADD     [BX+SI],AL
0116 0000      ADD     [BX+SI],AL
0118 0000      ADD     [BX+SI],AL

1      8 9 A B C D E F
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0050 CD 21 CB 00 00 00 00 00

2      F 0 1 2 3 4 5 6 7 8 9 A B C D E
DS:FFFF 00 CD 20 FF 9F 00 EA FF  FF AD DE 1B 05 C5 06 00  .= f.Ω  ;|..+.
DS:000F 00 18 01 10 01 18 01 92  01 01 01 01 00 02 FF FF  .....f  .....
DS:001F FF FF FF FF FF FF FF FF  FF FF FF FF FF EB 19 E6  δ.μ
DS:002F 11 A2 01 14 00 18 00 F5  19 FF FF FF FF 00 00 00  .ó.....J  .  ...
DS:003F 00 05 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri
```

## Step : 2

As you can see below after pressing F1 on the UBUNTU LINUX ( QWERTY KEYBOARD ) , The instruction of MEMORY ADDRESS ( 0100 ) runs and moves the value 5 into AX ( ACCUMULATOR REGISTER ).....

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0005 SI 0000 CS 19F5 IP 0103 Stack +0 0000 Flags 7200  
BX 0000 DI 0000 DS 19F5 +2 20CD  
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF  
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0

CMD >

1

8 9 A B C D E F

DS:0008 AD DE 1B 05 C5 06 00 00  
DS:0010 18 01 10 01 18 01 92 01  
DS:0018 01 01 01 00 02 FF FF FF  
DS:0020 FF FF FF FF FF FF FF FF  
DS:0028 FF FF FF FF EB 19 E6 11  
DS:0030 A2 01 14 00 18 00 F5 19  
DS:0038 FF FF FF FF 00 00 00 00  
DS:0040 05 00 00 00 00 00 00 00  
DS:0048 00 00 00 00 00 00 00 00  
DS:0050 CD 21 CB 00 00 00 00 00

2

F 0 1 2 3 4 5 6 7 8 9 A B C D E

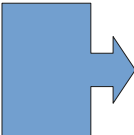
DS:FFFF 00 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00  
DS:000F 00 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF  
DS:001F FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6  
DS:002F 11 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00  
DS:003F 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00

. = f.Ω ÷ |..†..  
.....ff .....  
δ.μ  
.ó.....J . ...  
.....

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

Step : 3

As you can see below after pressing F1 on the UBUNTU LINUX ( QWERTY KEYBOARD ) , The instruction of MEMORY ADDRESS ( 0103 ) runs and moves the value 10 into BX ( BASE REGISTER ).....



```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0005 SI 0000 CS 19F5 IP 0106 Stack +0 0000 Flags 7200
BX 000A DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0

CMD >

0103 BB0A00 MOV BX,000A
0106 01D8 ADD AX,BX
0108 BB0F00 MOV BX,000F
010B 01D8 ADD AX,BX
010D B8004C MOV AX,4C00
0110 CD21 INT 21
0112 0000 ADD [BX+SI],AL
0114 0000 ADD [BX+SI],AL
0116 0000 ADD [BX+SI],AL

1 8 9 A B C D E F
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0050 CD 21 CB 00 00 00 00 00

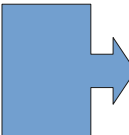
2 F 0 1 2 3 4 5 6 7 8 9 A B C D E
DS:FFFF 00 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00
DS:000F 00 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF
DS:001F FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6
DS:002F 11 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00
DS:003F 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

```

# Step : 4

As you can see below after pressing F1 on the UBUNTU LINUX ( QWERTY KEYBOARD ) , The instruction of MEMORY ADDRESS ( 0106 ) runs and adds the values of AX ( ACCUMULATOR REGISTER ) AND BX ( BASE REGISTER ) . Stores the result into AX .....



```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 000F SI 0000 CS 19F5 IP 0108 Stack +0 0000 Flags 7204
BX 000A DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 1 0

CMD >

0106 01D8 ADD AX,BX
0108 B80F00 MOV BX,000F
010B 01D8 ADD AX,BX
010D B8004C MOV AX,4C00
0110 CD21 INT 21
0112 0000 ADD [BX+SI],AL
0114 0000 ADD [BX+SI],AL
0116 0000 ADD [BX+SI],AL
0118 0000 ADD [BX+SI],AL

1 8 9 A B C D E F
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0050 CD 21 CB 00 00 00 00 00

2 F 0 1 2 3 4 5 6 7 8 9 A B C D E
DS:FFFF 00 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00
DS:000F 00 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF
DS:001F FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6
DS:002F 11 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00
DS:003F 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00

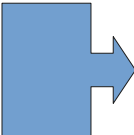
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

```

# Step : 5

As you can see below after pressing F1 on the UBUNTU LINUX ( QWERTY KEYBOARD ) , The instruction of MEMORY ADDRESS ( 0108 ) runs and moves the value 15 into BX ( BASE REGISTER ) .....





```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 000F SI 0000 CS 19F5 IP 010B Stack +0 0000 Flags 7204
BX 000F DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 1 0

CMD >

0108 BB0F00 MOV BX,000F
010B 01D8 ADD AX,BX
010D B804C MOV AX,4C00
0110 CD21 INT 21
0112 0000 ADD [BX+SI],AL
0114 0000 ADD [BX+SI],AL
0116 0000 ADD [BX+SI],AL
0118 0000 ADD [BX+SI],AL
011A 0000 ADD [BX+SI],AL

1 DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0050 CD 21 CB 00 00 00 00 00

2 F 0 1 2 3 4 5 6 7 8 9 A B C D E
DS:FFFF 00 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00
DS:000F 00 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF
DS:001F FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6
DS:002F 11 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00
DS:003F 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00

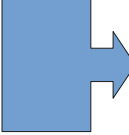
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

```

# Step : 6

As you can see below after pressing F1 on the UBUNTU LINUX ( QWERTY KEYBOARD ) , The instruction of MEMORY ADDRESS ( 010B ) runs and adds the value of AX ( ACCUMULATOR REGISTER ) AND BX ( BASE REGISTER ).Stores it into AX( ACCUMULATOR REGISTER .

.....



```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 001E SI 0000 CS 19F5 IP 010D Stack +0 0000 Flags 7214
BX 000F DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 1 1 0

CMD >

010B 01D8 ADD AX,BX
010D B8004C MOV AX,4C00
0110 CD21 INT 21
0112 0000 ADD [BX+SI],AL
0114 0000 ADD [BX+SI],AL
0116 0000 ADD [BX+SI],AL
0118 0000 ADD [BX+SI],AL
011A 0000 ADD [BX+SI],AL
011C 0000 ADD [BX+SI],AL

1
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0050 CD 21 CB 00 00 00 00 00

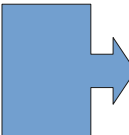
2
F 0 1 2 3 4 5 6 7 8 9 A B C D E
DS:FFFF 00 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00
DS:000F 00 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF
DS:001F FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6
DS:002F 11 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00
DS:003F 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

```

# Step : 7

As you can see below after pressing F1 on the UBUNTU LINUX ( QWERTY KEYBOARD ) , The instruction of MEMORY ADDRESS ( 010D ) runs and MOVES the value “4C00” ( which is an interrupt DEFINED by INTEL )...



```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 4C00 SI 0000 CS 19F5 IP 0110 Stack +0 0000 Flags 7214
BX 000F DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 1 1 0

CMD >

010D BB004C MOV AX,4C00
0110 CD21 INT 21
0112 0000 ADD [BX+SI],AL
0114 0000 ADD [BX+SI],AL
0116 0000 ADD [BX+SI],AL
0118 0000 ADD [BX+SI],AL
011A 0000 ADD [BX+SI],AL
011C 0000 ADD [BX+SI],AL
011E 0000 ADD [BX+SI],AL

1 DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0050 CD 21 CB 00 00 00 00 00

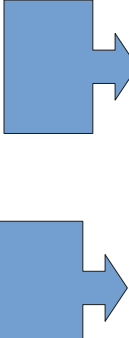
2 F 0 1 2 3 4 5 6 7 8 9 A B C D E
DS:FFFF 00 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00
DS:000F 00 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF
DS:001F FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6
DS:002F 11 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00
DS:003F 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

```

# Step : 8

As you can see below after pressing F1 on the UBUNTU LINUX ( QWERTY KEYBOARD ) , The instruction of MEMORY ADDRESS ( 0110 ) runs and CHECKS the most recent value in the AX ( ACCUMULATOR REGISTER ) register in which there is the 4C00 instruction ( which is an interrupt DEFINED by INTEL )...



```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 4C00 SI 0000 CS F000 IP 14A0 Stack +0 42BD Flags 7014
BX 000F DI 0000 DS 19F5 +2 06C5
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 7014 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF2 SS 19F5 FS 19F5 +6 0112 0 0 0 0 0 1 1 0

CMD >

0110 CD21 INT 21
14A0 FB STI
14A1 FE DB FE
14A2 3825 CMP [DI],AH
14A4 00CF ADD BH,CL
14A6 CB RET Far
14A7 51 PUSH CX
14A8 B94001 MOV CX,0140
14AB E2FE LOOP 14AB

1 DS:0008 AD DE 1B 05 C5 06 00 00
2 DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0050 CD 21 CB 00 00 00 00 00

F 0 1 2 3 4 5 6 7 8 9 A B C D E
DS:FFFF 00 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00
DS:000F 00 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF
DS:001F FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6
DS:002F 11 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00
DS:003F 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

```

## Step : 9

As you can see below after pressing F1 on the UBUNTU LINUX ( QWERTY KEYBOARD ) , The instruction of MEMORY ADDRESS ( 0110 ) runs and program TERMINATES

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0000	SI 0000	CS 19F5	IP 0100	Stack +0 0000	Flags 7202
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 0 0 0 0

CMD >

Program terminated OK

0100	B80500	MOV	AX,0005
0103	BB0A00	MOV	BX,000A
0106	01D8	ADD	AX,BX
0108	BB0F00	MOV	BX,000F
010B	01D8	ADD	AX,BX
010D	B8004C	MOV	AX,4C00
0110	CD21	INT	21
0112	0000	ADD	[BX+SI],AL

1		8	9	A	B	C	D	E	F
DS:0008	AD	DE	1B	05	C5	06	00	00	
DS:0010	18	01	10	01	18	01	92	01	
DS:0018	01	01	01	00	02	FF	FF	FF	
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	
DS:0028	FF	FF	FF	FF	EB	19	E4	11	
DS:0030	A2	01	14	00	18	00	F5	19	
DS:0038	FF	FF	FF	FF	00	00	00	00	
DS:0040	05	00	00	00	00	00	00	00	
DS:0048	00	00	00	00	00	00	00	00	
DS:0050	CD	21	CB	00	00	00	00	00	

2		F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
DS:FFFF	00	CD	20	FF	9F	00	EA	FF	FF	AD	DE	1B	05	C5	06	00	
DS:000F	00	18	01	10	01	18	01	92	01	01	01	01	00	02	FF	FF	
DS:001F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	E4	
DS:002F	11	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	
DS:003F	00	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

.	=	f.	Ω		;		..	+	..
.....	ft								
								δ.	Σ
.	ó	.....	J						
.....									

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri