

Princess Sumaya University for Technology Computer Engineering Department 22440: Microprocessor Lab

MASM Tutorial

Follow this tutorial step by step:

• You can use almost any text editor to create an assembly program. In this example, we will use Microsoft's EDIT. Type "edit example1.asm" on the command prompt and enter the text of the program.

Save the file by "Alt-F" and "Alt-S". Exit "Alt-F" and "Alt-X"



• Compile and link the assembly file by issuing "ml /Zi example1.asm"



- Now let us start and configure the Code View debugger. Type "cv example1.exe" at the command prompt. Enter "Alt-W" and make sure that you have the following windows on the screen:
 - \circ source1
 - o registers
 - o memory1

Press "Alt-F5" to arrange the windows on the screen.

C:\WINDOWS\System32\cmd.exe - cv examp	le1.exe _ 🗆 🗙
File Edit Search Bun Data Optio	ons Calls Indows Telp
[3] source1 CS:IP exam	ple1.asmI=[7]regist ==↓↑
9: MOU AX, Ø	;Clear register AX (AX=0 EAX = 00000000t
09DD:0000 B8DF09 MOV A	K,09DF EBX = 00000000
09DD:0003 8ED8 MOV D	S,AX ECX = 0000000
09DD:0005 8CD3 MOV B	K,SS EDX = 0000000
09DD:0007 2BD8 SUB B	K,AX ESP = 00000000
09DD:0009 C1E304 SHL B	$K_{*}04 = 0000000000000000000000000000000000$
09DD:000C 8ED0 MOV S	S, AX = 00000000
UYDD:UUUE U3E3 ADD S.	EDI = UUUUUUUU
0ADD:0010 R80000 WOA US	LOUDU DS = UYCD
10: MOV AL, VAR1	;Copy value inside memor ES = 09CD
NAND:NNI3 HNNCNN WOA H	L, BYIE PIR LUUUCI
11: 40. MOU DY OFFICET HADD	; into the register HL $GS = 0000$
12: NOV BA, UFFSEI VHKZ	r_{ABGD}
03 0010 00000 0000 000 D	5,000D 55 - 07DD ETP - 00000000
r-[5] memory1 b DS	:0 EFL = 00000200
09CD:0000 CD 20 FF 9F 00 9A F0 FE 1	D FØ 96 Ø2 = f.Ü≡∎+=û@ NV UP EI PL
09CD:000C D4 07 AB 03 D4 07 56 01 0	F 04 83 09 ⊧•½v⊧•U⊕x+âo NZ NA PO NC 🕺
09CD:0018 01 01 01 00 02 01 01 03 F	FFFFFFF GEE.CEEV
09CD:0024 FF FF FF FF FF FF FF FF FF 9	4 09 E8 36 öc⊉6
09CD:0030 0E D1 14 00 18 00 CD 09 F	FFFFFFFFff.t.=0
(Turse) (Ctan) (Ct) (Often Detune) (
(Trace/ Vaceb/ VGO/ (Hiter Return) ()	cau-uance1/

• Set the options through "**Alt-O**" -> **Preferences**. Set the options as shown below and click "OK".

	WS\System32\cmd.exe - cv example1.exe	- 🗆 ×
File Edi 9: 99DD:0000 99DD:0003 99DD:0007 99DD:0007 99DD:0007 99DD:0007 09DD:0007 09DD:0002 09DD:0010 10: 99DD:0013 11:	t Search Run Data Options Calls Windows Help Source1 CS:IP example1.asm MOU AX, 0 ESBDF09 MOU AX,09DF SED8 MOU DS,AX Preferences Scroll Bars IX1 orizontal IX1 orizontal IX1 ertical Display Radix: [Decimal·····]]	regist =:↓↑ = 00000000 = 00000000 00000000 00000000
12: 09DD:0016 [5] 09CD:0000 09CD:000C 09CD:000C 09CD:0024 09CD:0030 <f1=help></f1=help>	Imate Speed: IMedium IXI Case Sensitive < Colors> < Cancel>< OK > < Help> 01 01 01 00 02 01 01 03 FF FF FF FF 0000.0000 0000 FF	99DD 99DD 900000000 PEIPL 9000200 PEIPL

• Set Source 1 window options as shown below through "Alt-O" -> "Source 1 window"



• And the memory window options through "Alt-O" - > "Memory 1 window"



The configuration is now complete.

• Let us examine the **cv** program.

/ "	ine # from the source file Instructions in memory Instructions	uction nonics
C:\WP\DOWS\System File Edit Searc	3 Acmd.exe - cv example1.exe Run Data Options Calls Vindows Telp ource1 CS:IP example1.asn	- 🗆 ×
1:	EL SMALL ;One data and one code regments 1 EAX = 04 A ;Start of the data segrent EBX = 04 33H ;Allocate memory for variables ECX = 04 0101H 0AAAA5555H EDX = 04 0AAAA5555H EBP = 04 EBP = 04 ;Enable 32-bit ESI = 04 ESI = 04 RTUP ;The program starts here EDI = 04 AX, 0 :Clean register AX (AX=0) ES = 05 MOU DS, AX FS = 04	1999999 19999999 19999999 19999999 1999999
09DD:0007 2DB8 09DD:0007 2DB8 09DD:0009 CIE304 49CD 000C 14 07 A1 09CD 000C 14 07 A1 09CD 000C 14 07 A1 09CD 0018 11 01 05 09CD 0024 7F FF FF	NOV BX,3X State State	DD DD 0000000 000200 I PL O NC
Trace> (Step> <go Offset Value of the DS</go 	After Return> <p3=s1 pmt=""> <sn+13=m1 pmt=""> IN Remory content shown as byte-size hexadecimal numbers but shown in ASCII</sn+13=m1></p3=s1>	DEC Əgister



- Step through the program and observe the execution of every instruction.
 - Press "**F10**".
 - The debugger shows execution of the first line of the initialization code.
 - Press "**F10**" multiple times until the instruction "MOV AX, 0" is highlighted. This is the first instruction of your program.

C:\WINDO	WS\System3	2 \cmd.e	ke - cve	xample1.e	хе			- 🗆 ×
File Ed	it Search	Run	Data	Options	Calls	: lindows	lelp	
	S NOT	ourcel	CSTP	example1	.asm =	AU		[17]regist
7: 0000-0000	MUU H	х, Ю	AU.	AU 90	lear r	egister HX	CHX=0	EHX = 000009DF
0400:0000	BSDF09		00	HX,07	DF			EBX = UUUUFFEU
0900:0003	8ED8			DS, HX				ECX = 000000000
0700:0005	8603		NON CONTRACTOR	BA, 55				EDX = 00000000000000000000000000000000000
0900:0007	ZBUS	2	UB	BX,HX				ESP = 31515151313135
0700:0007	GIE304	2	HL .	BX,04				EBP = 000000000
09DD:000C	8ED0	, in the second s		33,HA				E21 = 000000000
09DD:000E	UJEJ	H	עע	5P, BA	00			FDI = 00000000000000000000000000000000000
0400:0010	BRANNON		IUV	нх,00	ออ	1		DS = DYDF
10:	MUU H.	ь, онкі	AU	AT DU	opy va	lue inside	memor	ES = 09CD
MADD: MAT3	нөөсөө	l III	100	HL, BY	TE PIP	เโดดดด่า		F2 = 0000
11:	MOU D		TT HAT	10 11	nto tr	e register	HL .	C2 = N0NN
12:	DDODOO B	X, UFFS	EI VHI		Lace o	offset of VI	IKZ IN	55 = 5905
NADD: NOIP	ввалаа		00	BX,00	ND N			GS = MYDD
-							→-1	EIP = SUSSES
	DO EZ 04	me	mory1	D DS:0 -	00.04	IL AAC (EFL = 515151532812
09DF:0000	78 56 34	12 66	HJ OF	00 B4 40	CD Z1	XV4Ifu@.	L=i	NU UP EI NG
UYDF: UUUC	33 01 01	55 55	HH HH	4E 4E 42	30 37	JEEUUTIN	ивыа	NZ NH PU IU
07DF:0018	98 02 00	00 00	00 00	01 01 00	43 56	yaa	9.C0	
07DF:0024	01 00 00	00 00	00 00	00 20 00	00 00			
07DE:0030	06 65 78	61 6D	70 60	65 31 ZE	6F 62	: ¥example.	0.00	
<trace> <</trace>	Step> <go></go>	<after </after 	Retur	•n> <f3=s< td=""><td>1 Fmt></td><td>> <sh+f3=m1< td=""><td>Fmt></td><td>INS DEC</td></sh+f3=m1<></td></f3=s<>	1 Fmt>	> <sh+f3=m1< td=""><td>Fmt></td><td>INS DEC</td></sh+f3=m1<>	Fmt>	INS DEC

Observe the value in the register EAX. Register AX contains the number 09DFH.

📾 C:\WINDOWS\System32\cmd.exe - cv example1.exe	- 🗆 ×
File Edit Search Run Data Options Calls Windows Help	- i7lregist
9: NOU AX, 0 ;Clear register AX (AX=0) 99D:0000 B8DF09 MOU AX, 09DF 09DD:0005 8CD3 MOU DS, AX 09DD:0007 2BD8 SUB BX, AX 09DD:0006 8ED0 MOU DS, AX 09DD:0007 2BD8 SUB BX, AX 09DD:0006 8ED0 MOU SS, AX 09DD:0006 8ED0 MOU SS, AX 09DD:0006 8E00 MOU SS, AX 09DD:0010 B80000 MOU AX, 0000 10: MOU AL, UAR1 ;Copy value inside memor 09DD:0013 A00C00 MOU AL, BYTE PTR [000C] 11: ;into the register AL 12: MOU BX, OFFSET UAR2 ;Place offset of UAR2 in 09DD:0016 BB0D00 MOU BX,000D 14	EAX = 000009DF ECX = 00000000 EDX = 00000000 ESP = <u>353399E5</u> EBP = 00000000 ESI = 00000000 EDI = 00000000 DS = 09DF ES = 0900 GS = 0000 SS = <u>59DF</u> CS = 09DD EIP = <u>3533282</u> NU UP EI NG NZ NA PO <u>TC</u>
<trace> <step> <go> <after return=""> <f3=s1 fmt=""> <sh+f3=m1 fmt=""></sh+f3=m1></f3=s1></after></go></step></trace>	INS DEC

• Now press "**F10**". The debugger will execute the highlighted instruction. Note the change in the content of EAX and the fact that the register has been highlighted by the debugger, indicating the change.

C:\WINDOWS\System32\cmd.exe - cv example1.exe	- 🗆 ×
File Edit Search Fun Data Options Calls Windows Help	
I=[3] source1 CS:IP example1.asm	lî - l'/ iregist
9: MOU AX, Ø ;Clear register AX (AX=	0 1 EAX = <u>313131313131</u>
09DD:0000 B8DF09 MOV AX,09DF	EBH COCOTTEC
09DD:0003 8ED8 MOU DS,AX	ECX = 00000000
09DD:0005 8CD3 MOU BX,SS	EDX = 00000000
M9DD:0007 2BD8 SUB BX,AX	ESP = 0000FFE0
09DD:0009 C1E304 SHL B8,04	EBL = NNNNNNN
AADD: AAAC SEDA WOO SS'AX	ESI = 000000000
BYDD: GOUE 03E3 HDD SP, BX	EDI = 00000000
AADD:0010 B80000 HOO HX,0000	DS = 09DF
10: NOV HL, VHKI ;CODY VALUE INSIDE memo	$\mathbf{F}_{\mathbf{E}} = 07\mathbf{C}\mathbf{D}$
AADD-0013 H00C00 HOU HL, BYIE FIR L000CI	FS = 0000
11: , into the register HL 12: Moll DY OPPET HOP2 ; Disco affect of HD2;	GS = 0000
ACD - ACC DADAGA MAL BY AGAD	CC = 07DT
- [5] memory() - 5, 08-6	FFL = 00003282
09DF:0000 28 56 34 12 66 A3 0F 00 B4 4C CD 21 vU4tfút -L=*	NU UP EL NG
09DF:000C 33 01 01 55 55 AA AA 4F 4F 42 30 39 300000-0000	NZ NA PO NC
09DF:0018 98 02 00 00 00 00 01 01 00 43 56 UB	
09DF:0024 01 00 00 00 00 00 00 2C 00 00 00 0	ds:000c
09DF:0030 0C 65 78 61 6D 70 6C 65 31 2E 6F 62	33
<pre> <trace> <step> <go> <after return=""> <f3=s1 fmt=""> <sh+f3=m1 fmt=""></sh+f3=m1></f3=s1></after></go></step></trace></pre>	INS DEC

The highlighting of the code window moved to the next instruction.

Note that the line of the source code "MOV AL, VAR1" became "MOV AL, [000C]" where 000CH is the actual offset of VAR1 in the data segment. You can check that this is true by checking the contents of memory location DS:000CH in the memory window.

• Now execute this instruction by pressing "**F10**". The content of the register AL changes, taking the value from VAR1.

📾 C:\WINDOWS\System32\cmd.exe - cv example1.exe	- 🗆 ×
Image: Solucity of the sector of the sect	- C × EAX = <u>313131383</u> EBX = 0000FFC ECX = 00000F9 ECX = 00000603 ESP = 00000F7C0 EBP = 00000603 ESI = 00000100 ESI = 00000100 EDI = 00000100 EDI = 00000100 ESI = 09DF ES = 09CD FS = 0000 CS = 0000
17: 1000 EHA, 12:15078H (Load the number 12:34567 09DD:001E 66B878563412 MOU EAX,12:345678 18: :into the register EAX :into the register EAX 09DF:0000 28 55 34 12 66 A3 0F 00 B4 4C CD 21 xU4tfú*. L=! 09DF:0000 33 01 01 55 55 00 00 4E 4E 42 20 20 20000 - NMD00 09DF:0018 76 02 00 00 00 00 00 00 101 00 43 56 100	GS = 0900 SS = 09D) CS = 09D) EIP = <u>30930516</u> EFL = 00103282 NU UP EI NG NU UP EI NG

• The next instruction is "MOV BX, OFFSET VAR2". VAR2 follows VAR1 in memory and has offset 000DH. This is the value that will be placed into the BX register upon execution of this instruction. Press "**F10**" to execute.

C:\WINDOWS\System32\cmd.exe - cv example1.exe	- 🗆 ×
File Edit Search Bun Data Options Calls Windows Help	e anna anna
I=[3] source 00 IP cyample1.asm	[7]regist
12: MOU BX, OFFSET VAR2 :Place offset of VAR2 int	EAX = UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
RADD: ONIP BRODON LDAT VISA BY ODOD	EBX = 5550155500
13: GODD-GAIG 9967 MAL GCODY VALUE FROM THE PEG	ECX = 00000000
14.	EDA = 00000000
15: MOII [RX+1] AL :Conv using the per	FRP = 00000000000000000000000000000000000
09DD:001B 884701 MOU BYTE PTR [BX+01].AL	ESI = 00000000
16: the memory location poi	EDI = 00000000
17: MOU EAX, 12345678H ;Load the number 1234567	DS = 09DF
09DD:001E 66B878563412 MOV EAX,12345678	ES = 09CD
18: ;into the register EAX	FS = 0000
19: MOU VAR3, EAX ;Copy value from the reg	GS = 0000
MOU DWORD PIR LUUUFI, EAX	SS = U9DF
20: the memory location VHR	$CS = \underline{M}\overline{M}\overline{M}$
	EIP = 319191919191919191919191919191919191919
151 MENDER 100 78 56 34 12 66 03 0F 00 84 4C CD 21 VIAttim 11-1	NIL IIP ET NC
99DF-990C 33 01 01 55 55 44 46 47 47 40 39 300 111	NZ NA PO NC
09DF:0018 98 72 77 00 00 00 00 01 01 00 43 56 UB	
09DF:0024 01 00 00 00 00 00 00 2C 00 00 00 0	ds:000d
09DF:0030 0C 65 78 61 6D 70 6C 65 31 2E 6F 62	01
<pre><trace> <step> <go> <after return=""> <f3=s1 fmt=""> <sh+f3=m1 fmt=""></sh+f3=m1></f3=s1></after></go></step></trace></pre>	INS DEC

• The following instruction "MOV [BX], AL" will copy the content of AL into the memory location pointed to by BX within the data segment. After the previous instruction BX contains the offset of the first byte of VAR2 or 000DH. That is where the data from AL will appear. Press "F10" to execute. Note the debugger also highlighted changes in the data window.

C:\WINDOWS\System32	2\cmd.exe - cv example1.exe	- 🗆 ×
File Edit Search	Fun Data Options Calls Windows Help	attender and and a
I=[3] so	ource1 CS:IP example1.asm	-[7]regist 🕋
13:MOV []	BX], AL ;Copy value from the reg <mark>f</mark>]	EAX = 000000333
09DD:0019 8807	MOV BYTE PTR [BX],AL	EBX = 0699000D
14:	;the memory location poi	ЕСХ ИНИИИИИИ
	BX+1],AL ;Copy value from the reg [$\mathbf{D}\mathbf{X} = \mathbf{D}\mathbf{D}\mathbf{D}\mathbf{D}\mathbf{D}\mathbf{D}\mathbf{D}$
09DD:001B 884701	MUV BYLE PIK LBX+01 J, HL	ESP = 0000FFE0
10- 10-	The memory location points	EBP = 00000000
09DD-001E 6689785634	10, 123430700 , LUGU CHE HUMBER 1234307 1	EDI = 00000000
18.	into the veristev FOX	DC = MODE
19: MOU U	AR3 FAX :Conversion the register	FS = M9CD
09DD:0024 66830F00	MOU DUORD PTR [000F] FAX	FS = 0000
20:	the memory location VAR	GS = 0000
21: .EXIT	;Exit to DOS	SS = 09DF
22: END		CS = 09DD
L ,	Li€	EIP = 0000001B
r-[5]	memory1 b DS:0	EFL = 00003282
09DF:0000 78 <u>56 3</u>	12 66 A3 ØF ØØ B4 4C CD 21 x⊻4‡fú≉.¦L=!	NU UP EI NG
09DF:000C 33 38 01	55 55 AA AA 4E 4E 42 30 39 30000000 NNB09	NZ NA PO NC
09DF:0018 98 02 00	00 00 00 00 01 01 00 43 56 ÿ e ⊙⊙.CV	
09DF:0024 01 00 00	NN NN NN NN NN SC NN NN NN ⊡	ds:000e
09DF:0030 0C 65 78	61 6D 70 6C 65 31 2E 6F 62 Yexamplel.ob	01
<trace> <step> <go></go></step></trace>	<pre><after return=""> <f3=s1 fmt=""> <sh+f3=m1 fmt=""></sh+f3=m1></f3=s1></after></pre>	INS DEC

• Instruction "MOV [BX+1], AL" copies the contents of register AL into the memory location with offset equal whatever the number is in BX plus 1. In our case BX=000DH, then the offset is 000DH+0001H=000EH. This is the second byte of VAR2. Press "**F10**" to execute. Note the change in the memory contents.

C:\WINDOWS\System32\cmd.exe - cv example1.exe	- 🗆 ×
File Edit Search Run Data Options Calls Windows Help	A REAL STREET
I=[3] source1 CS:IP example1.asm↓↑	[-[7]regist -
13: MOU [BX], AL ;Copy value from the reg <mark>f</mark>	EAX = 00000033
09DD:0019 8807 MOV BYTE PTR [BX],AL	EBX = 000000000
14: ;the memory location poi	ECX = 00000000
15: MOV [BX+1],AL ;Copy value from the reg	EDX = 00000000
09DD:001B 884701 MOV BYTE PTR [BX+01],AL	$\mathbf{F}\mathbf{S}\mathbf{F} = 0000\mathbf{F}\mathbf{F}\mathbf{E}0$
16: ;the memory location poi	EBP = 00000000
17: MOU EAX, 12345678H ;Load the number 1234567	ESI = 00000000
DYDD:001E 668878563412 MOV EAX,12345678	EDI = NONONONON
18: ;into the register EAX	DS = U9DF
19: GODY VHK3, EHX ; CODY Valge from the reg	ES = MYCD
DYDD:0024 66H30F00 MOV DWORD PDA L000FJ,EHX	FS = 0000
20: The memory location VHA	CC = 0000
20. END EXIT DUS	33 = 070F
ZZ- END	ETD - SISISISISTER
	EIF = 00000000000000000000000000000000000
101 PE-0000 78 56 24 12 56 02 0P 00 P4 40 CD 21 VII4+5/* JI-+	NIL IIP ET NC
GODE-GOOD 33 33 221 55 00 00 4F 4F 43 30 32 32 32 111	NZ NO PO NC
	112 111 10 110
M9DF: M030 MC 65 28 61 6D 20 6C 65 31 2E 6F 62 Pexample1.ob	
	ليستعمل
<pre><trace> <step> <go> <after return=""> <f3=s1 fmt=""> <sh+f3=m1 fmt=""></sh+f3=m1></f3=s1></after></go></step></trace></pre>	INS DEC

• Instruction "MOV EAX, 12345678H" places the number 12345678H into register EAX. Press "**F10**" to execute.

C:\WINDOWS\System32\cmd.exe - cv example1.exe	_ 🗆 🗙
File Edit Search Run Data Options Calls Windows Welp	
-13] Sourcel CS:IP example1.asm	EOV - BEELETAD
ACT ACT AND TO THE ACT AND A COLOR TO THE PERIOD AND A COLOR TO THE ACT AND A COLOR AND A	EHA - 123451676
14: the memory location noi	= 00000000
15: MOU [RX+1] AL :CONU UNLY From the Port	FDX = 000000000
09DD:001B 884701 MOU BYTE PTR [BX+01].AL	ESP = 0000FFE0
16: the memory location poi	EBP = 00000000
17: MOU EAX, 12 3456781 , Load the number 1234567	ESI = 00000000
09DD:001E 66B878563412 MOU EAX,12345678	EDI = 00000000
18: ,into the register EAX	DS = 09DF
19: MOU VAR3, EAX ;Copy value from the reg	ES = 09CD
<u>0</u> 9DD:0024 66A30F00 MOV DWORD PTR [000F1,EAX	FS = 0000
20: ;the memory location VAR	GS = MMMM
21: EXII ;Exit to DOS	SS = U9DF
ZZ: END	CS = UYDD
	EIP = 33333333244 EPI = 000000000
100 DE-0000 28 56 34 12 66 03 0F 00 R4 40 CD 21 vil4tfix 41-+	NIL IIP FI NC
AGDE: 000C 33 33 33 55 55 40 40 4F 4F 42 30 39 333111 TNNE09	NZ NA PO NC
09DF:0024 01 00 00 00 00 00 00 2C 00 00 00 0	ds:000f
09DF:0030 0C 65 78 61 6D 70 6C 65 31 2E 6F 62	aaaa5555
<pre><trace> <step> <go> <after return=""> <f3=s1 fmt=""> <sh+f3=m1 fmt=""></sh+f3=m1></f3=s1></after></go></step></trace></pre>	INS DEC

• The instruction "MOV VAR3, EAX" became "MOV DWORD PTR [000F], EAX". VAR3 has been replaced by the actual offset (000FH) of VAR3 in the data memory. This instruction takes the contents of EAX and places them in the four consecutive bytes of memory (a 32-bit variable) starting at the offset 000FH. Press "**F10**" to execute.

🖎 C:\WINDOWS\System32\cmd.exe - cv example1.exe	- 🗆 ×
File Edit Search Run Data Options Calls Windows Help	
I=[3] source1 CS:IP example1.asmI↑	[7]rgist
09DD:0016 BB0D00 MOV BX,000D 1	EAX = 12345678
13:MOU [BX], AL;Copy value from the reg	EBX - 00000000
09DD:0019 8807 MOV BYTE PTR [BX],AL	$E_{0} = 00000000$
14: ;the memory location poi	EDX = 000000000
15: MOV [BX+1],AL ;Copy value from the ref	ESP = 0000FFE0
MADD: MAIR 8845/01 MOO BAIE LEX HOI I'U	EBP = 000000000
16: ;the memory location pol	F21 = NONONONON
12345678H ;Load the number 1234567	FDI = 000000000
09DD:001E 668878563412 MOV EHX.12345678	DS = DYDF
10: NOU HOD2 FOY ; Into the register LHX	ES = 0700
17- NOV VHAS, EHA , GODY VALUE FOM THE FEY	F5 = 0000
	GS = 0000
	CC = 07DT
	FIP = GIGIGIGIGIGIPIPI
	$FFI_{} = 00003282$
09DE:0000 78 56 34 49 66 00 9D 69 84 40 CD 21 vII4tfús IL=	NIL LIP ET NG
ASDE: BOAC 33 33 33 23 37 62 19 4F 4F 42 36 39 333712FNNBA9	NZ NA PO NC
09DF:0018 98 02 00 00 00 00 01 01 00 43 56 UB 00 CU	
19DF 1024 01 00 00 00 00 00 00 00 20 00 00 00	
09DF:0030 0C 65 78 61 6D 70 6C 65 31 2E 6F 62 Pexample1.ob	
<pre><trace> <step> <go> <after return=""> <esc=cance1></esc=cance1></after></go></step></trace></pre>	

• This was the last instruction of the user program. The remaining instructions are generated by the **.EXIT** directive and serve to terminate the program. Press **"F10"** multiple times until the process terminates.

C:\WINDOWS\Syste	em32\cmd.exe - cv example1.exe	- 🗆 ×
File Edit Sea =[3] 99DD:002A CD21 09DD:002C 3333	rch Run Data Options Calls Windows Help source1 CS:IP INT 21 XOR SI,WORD PTR [BP+DI]	E71regist EAX = 12344673 EBX = 00000000
09DD:002E 337856 09DD:0031 3412 09DD:0033 4E 09DD:0034 4E 09DD:0035 42	XOR DI,WORD PTR [BX+SI+56] XOR AL,12 DEC SI DEC SI INC DX	ECX = 0000000 EDX = 0000000 ESP = 0000FFE0 EBP = 0000000 ESI = 0000000
09DD:0036 3039 09DD:0038 98 09DD:0039 0200 09DD:003B 0000 09DD:003B 0000	Process 0x09CD terminated normally (120)	EDI = 00000000 DS = 09DF ES = 09CD FS = 0000 CS = 0000
09DD:003F 0101 09DD:0041 004356 ←	K →	SS = 09DF L CS = 09DD L EIP = <u>300000000000000000000000000000000000</u>
09DF:0000 78 56 09DF:000C 33 33 09DF:0018 98 02 09DF:0024 01 00 09DF:0030 0C 65	34 12 66 A3 ØF 00 B4 4C CD 21 xU4‡fú*{L=! 33 78 56 34 12 4E 4E 42 30 39 333xU4‡NNB09 00 00 00 00 01 01 00 43 56 jt 00 00 00 00 00 02 00	NU UP EI NG NZ NA PO NC
<trace> <step> <</step></trace>	Go> <after return=""> <f3=s1 fmt=""> <sh+f3=m1 fmt=""></sh+f3=m1></f3=s1></after>	INS DEC

Reference

http://www.intelligent-systems.info/classes/ee360/tutorial.htm