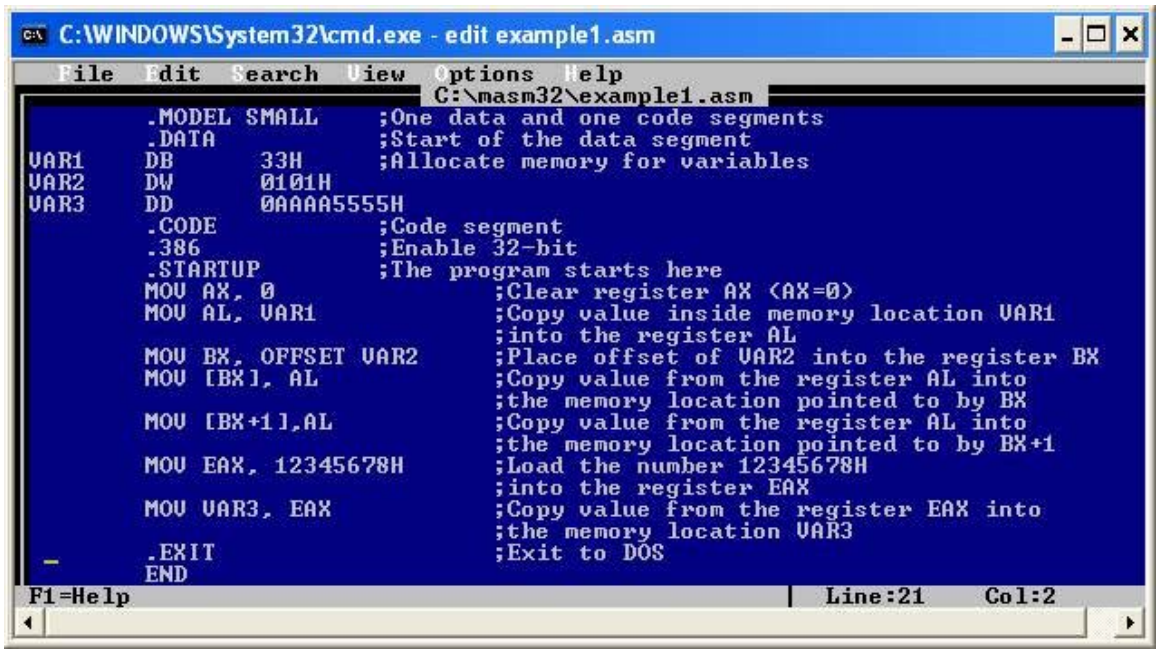
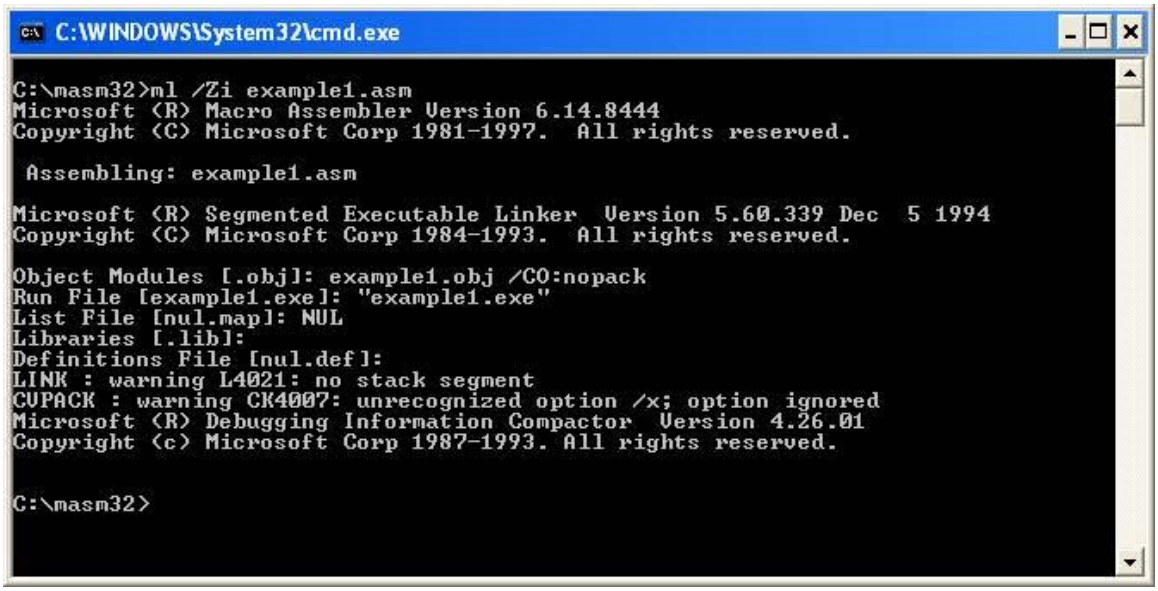
MASM Tutorial

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", "Alt+S". Exit "Alt-F", "Alt-X"

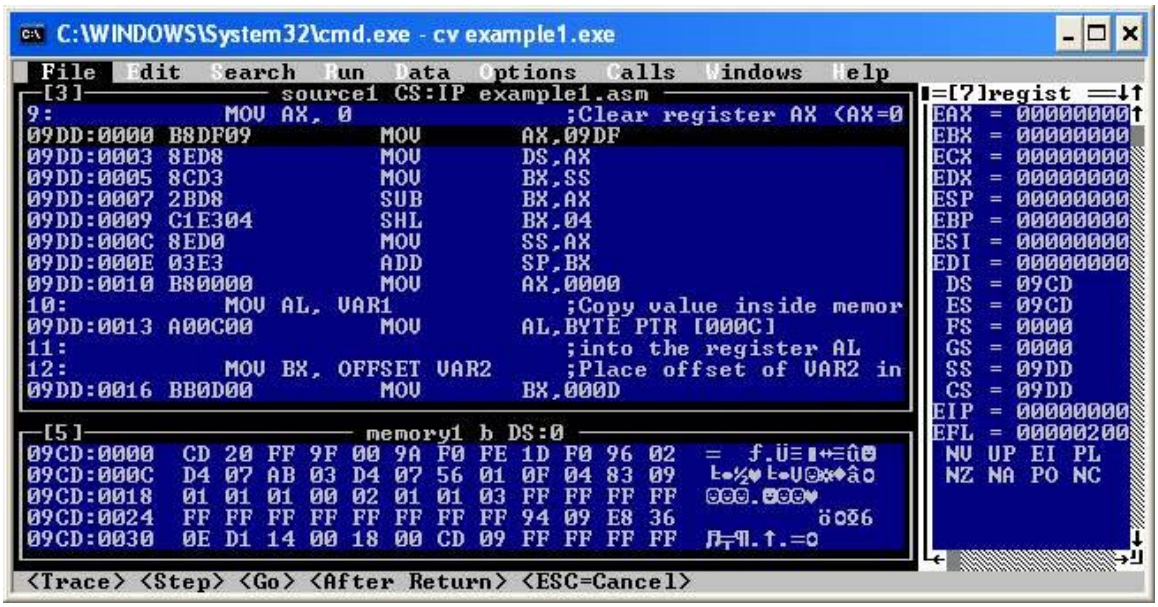


- Compile and link the assembly file by issuing "ml /Zi example1.asm"

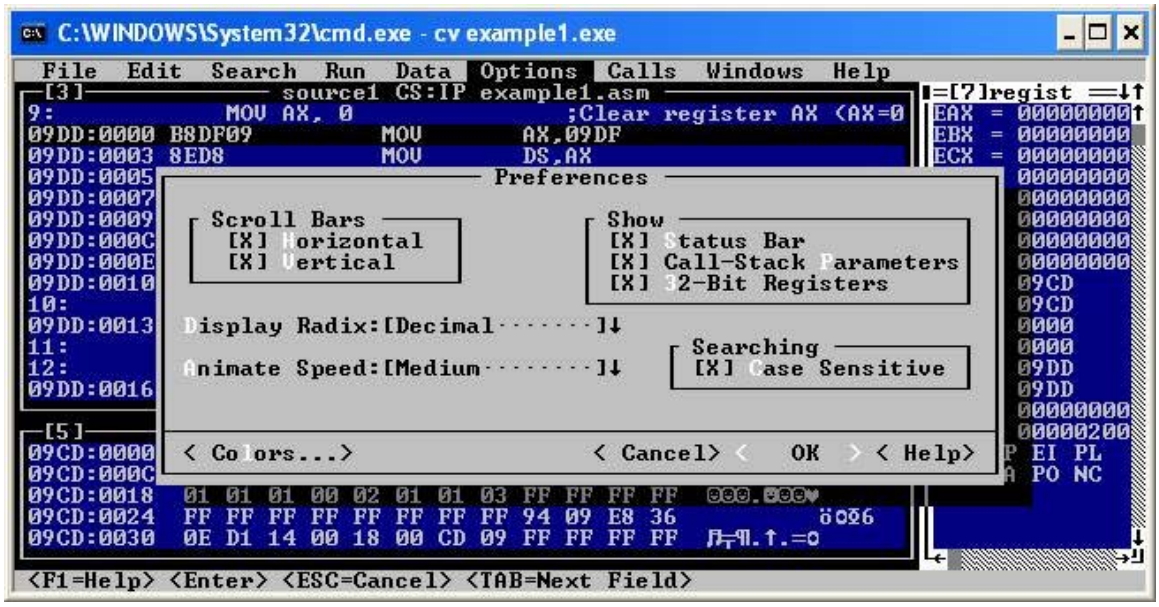


- Now let’s start and configure the Code View debugger.
  - Type “cv example1.exe’ at the command prompt.Enter “Alt-W” and make sure that you have the following windows on the screen:
    - Code1
    - Registers
    - Memory 1

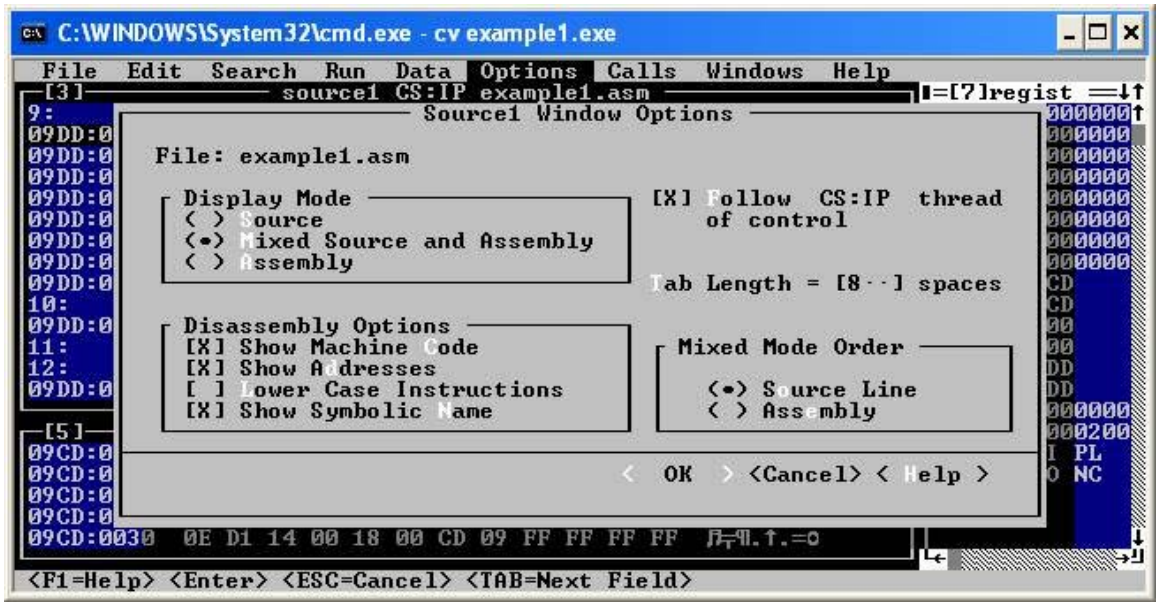
Press “Alt-F5” to arrange the windows on the screen.



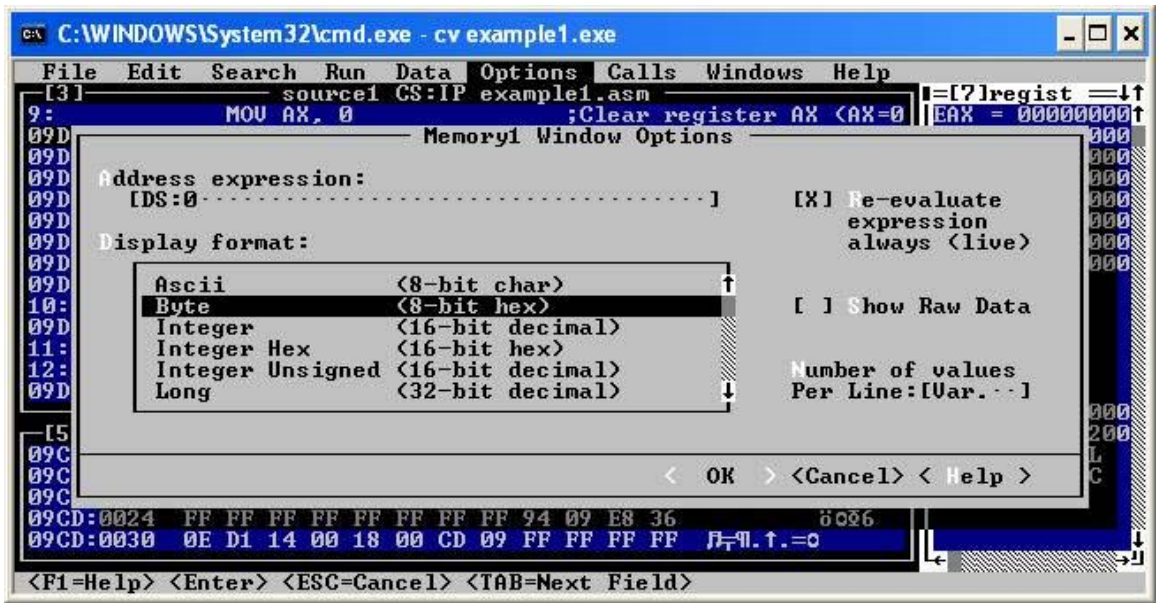
- Set the options. “Alt-O” -> Preferences. Set the options as shown and click “ok”.



- Again, “Alt-O” -> “Source 1 window”

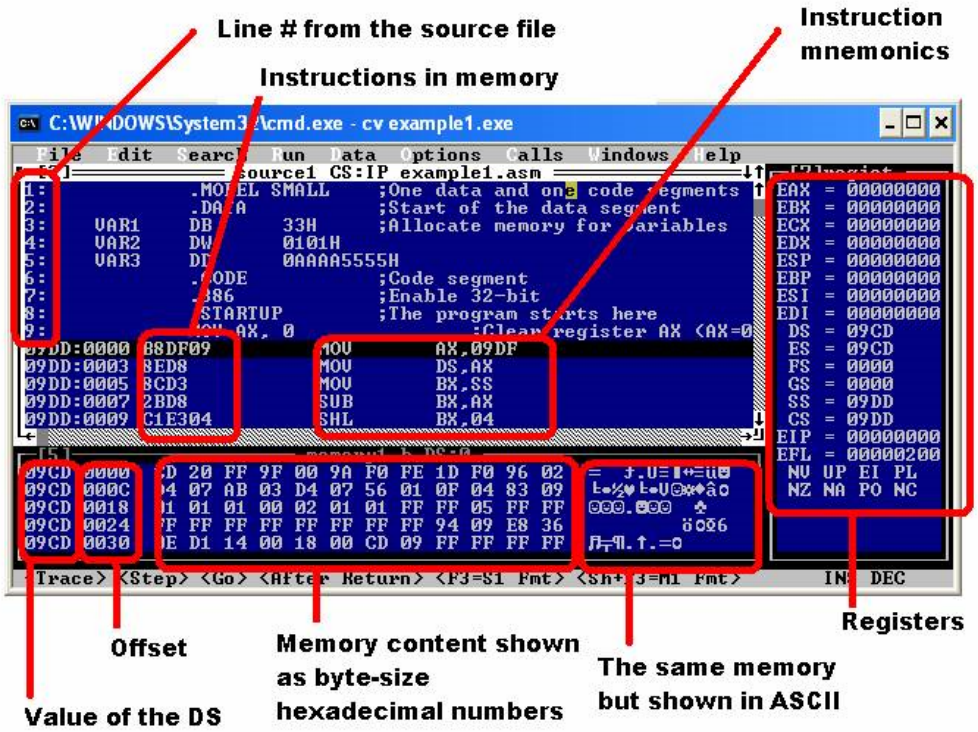


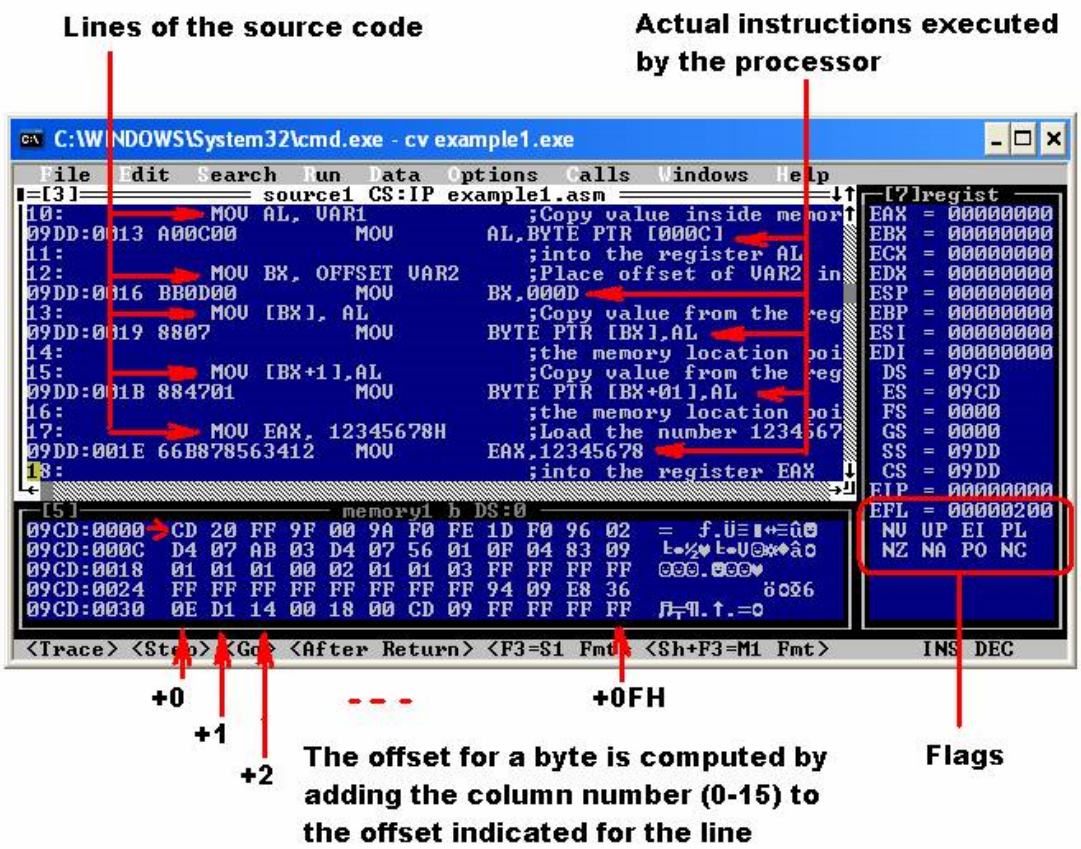
- “Alt-O” - >” Memory 1 window”

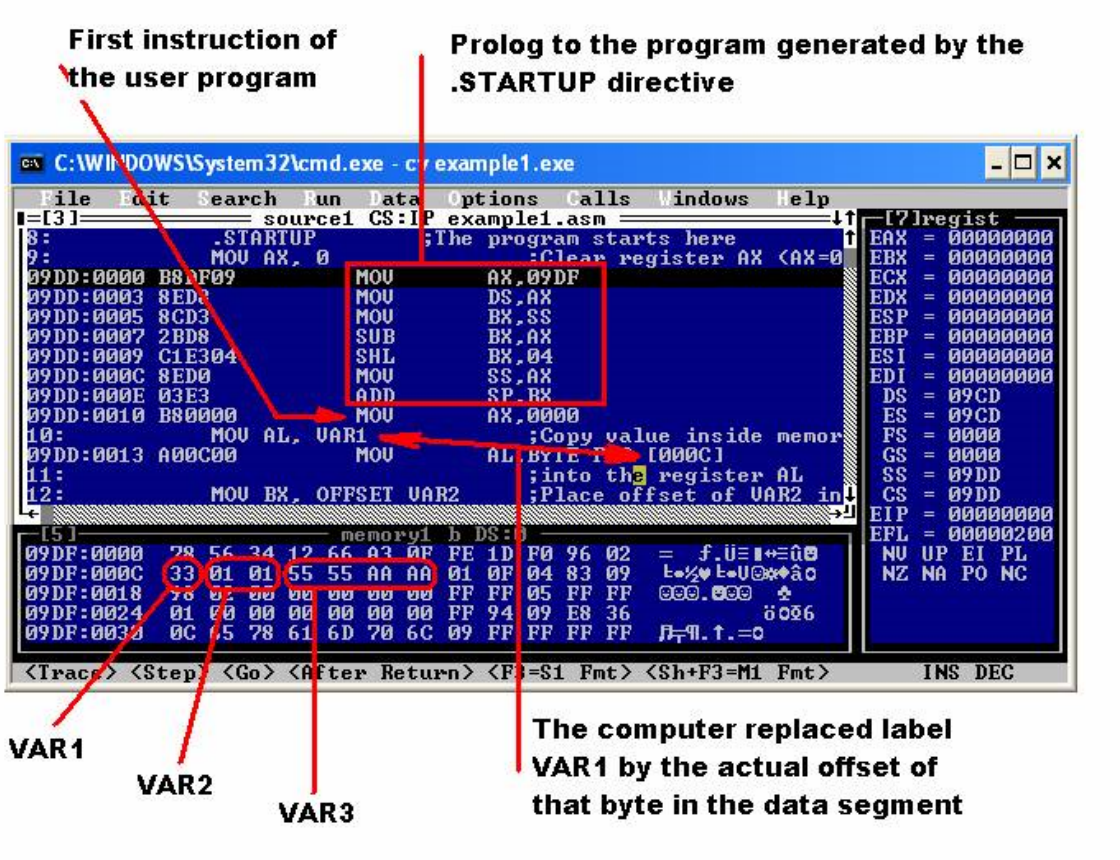


The configuration is now complete.

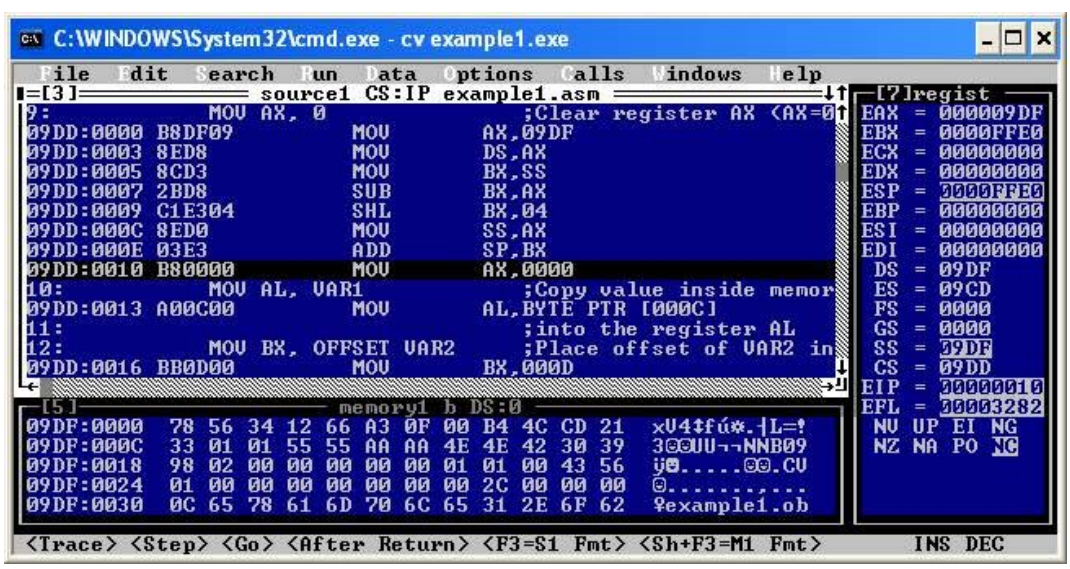
- Let’s look at the program.







- step through the program and observe execution of each instruction.
  - Press “F10”.
  - The debugger will show execution of the first line of the prolog.
  - Press “F10” until instruction “MOV AX,0” is highlighted. This is the first instruction of your program.



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

```
C:\WINDOWS\System32\cmd.exe - cv example1.exe
File  Edit  Search  Run  Data  Options  Calls  Windows  Help
[3]=========== source1 CS:IP example1.asm ==========
9:              MOV AX, 0                ;Clear register AX (AX=0
09DD:0000 B8DF09          MOV       AX,09DF
09DD:0003 8ED8            MOV       DS,AX
09DD:0005 8CD3            MOV       BX,SS
09DD:0007 2BD8            SUB       BX,AX
09DD:0009 C1E304          SHL       BX,04
09DD:000C 8ED0            MOV       SS,AX
09DD:000E 03E3            ADD       SP,BX
09DD:0010 B80000          MOV       AX,0000
10:             MOV AL, VAR1             ;Copy value inside memor
09DD:0013 A00C00          MOV       AL,BYTE PTR [000C]
11:                                      ;into the register AL
12:             MOV BX, OFFSET VAR2      ;Place offset of VAR2 in
09DD:0016 BB0D00          MOV       BX,000D
[5]=========== memory1 b DS:0 ==========
09DF:0000  78 56 34 12 66 A3 0F 00 B4 4C CD 21   xV4‡fú*.┤L=!
09DF:000C  33 01 01 55 55 AA AA 4E 4E 42 30 39   3☺☺UU¬¬NNB09
09DF:0018  98 02 00 00 00 00 00 01 01 00 43 56   ÿ♥.....☺☺.CV
09DF:0024  01 00 00 00 00 00 00 00 2C 00 00 00   ☺.......,...
09DF:0030  0C 65 78 61 6D 70 6C 65 31 2E 6F 62   ♀example1.ob
EAX = 000009DF
ECX = 00000000
EDX = 00000000
ESP = 0000FFE0
EBP = 00000000
ESI = 00000000
EDI = 00000000
DS = 09DF
ES = 09CD
FS = 0000
GS = 0000
SS = 09DF
CS = 09DD
EIP = 00000010
EFL = 00003282
NV UP EI NG
NZ NA PO NC
<Trace> <Step> <Go> <After Return> <F3=S1 Fmt> <Sh+F3=M1 Fmt>  INS DEC
```

Now press “F10”. The debugger will execute the highlighted instruction.

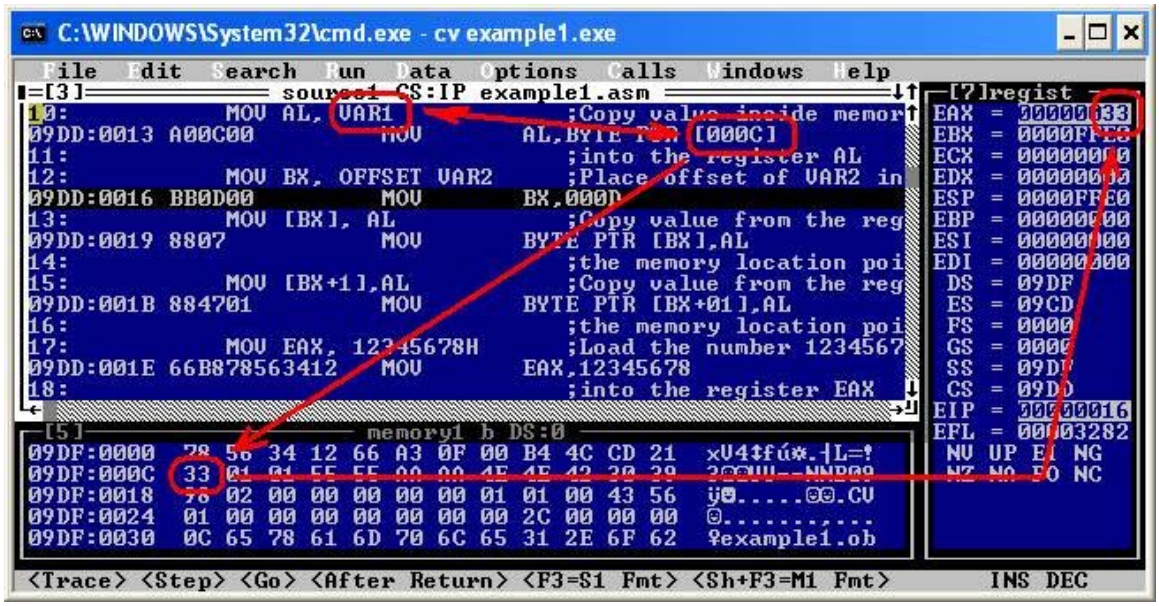
Note the change in the content of EAX and the fact that the register has been highlighted by the debugger, indicating the change.

```
C:\WINDOWS\System32\cmd.exe - cv example1.exe
File  Edit  Search  Run  Data  Options  Calls  Windows  Help
[3]=========== source1 CS:IP example1.asm ==========
9:              MOV AX, 0                ;Clear register AX (AX=0
09DD:0000 B8DF09          MOV       AX,09DF
09DD:0003 8ED8            MOV       DS,AX
09DD:0005 8CD3            MOV       BX,SS
09DD:0007 2BD8            SUB       BX,AX
09DD:0009 C1E304          SHL       BX,04
09DD:000C 8ED0            MOV       SS,AX
09DD:000E 03E3            ADD       SP,BX
09DD:0010 B80000          MOV       AX,0000
10:             MOV AL, VAR1             ;Copy value inside memor
09DD:0013 A00C00          MOV       AL,BYTE PTR [000C]
11:                                      ;into the register AL
12:             MOV BX, OFFSET VAR2      ;Place offset of VAR2 in
09DD:0016 BB0D00          MOV       BX,000D
[5]=========== memory1 b DS:0 ==========
09DF:0000  78 56 34 12 66 A3 0F 00 B4 4C CD 21   xV4‡fú*.┤L=!
09DF:000C  33 01 01 55 55 AA AA 4E 4E 42 30 39   3☺☺UU¬¬NNB09
09DF:0018  98 02 00 00 00 00 00 01 01 00 43 56   ÿ♥.....☺☺.CV
09DF:0024  01 00 00 00 00 00 00 00 2C 00 00 00   ☺.......,...
09DF:0030  0C 65 78 61 6D 70 6C 65 31 2E 6F 62   ♀example1.ob
EAX = 00000000
ECX = 00000000
EDX = 00000000
ESP = 0000FFE0
EBP = 00000000
ESI = 00000000
EDI = 00000000
DS = 09DF
ES = 09CD
FS = 0000
GS = 0000
SS = 09DF
CS = 09DD
EIP = 00000013
EFL = 00003282
NV UP EI NG
NZ NA PO NC
ds:000c
33
<Trace> <Step> <Go> <After Return> <F3=S1 Fmt> <Sh+F3=M1 Fmt>  INS DEC
```

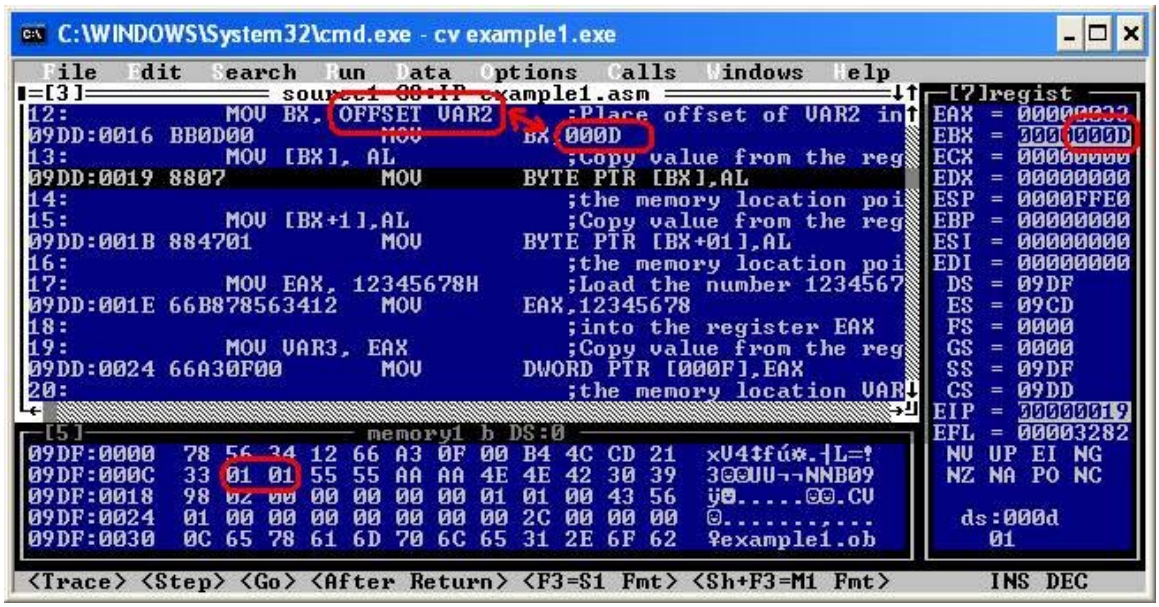
The highlighting the code window moved to the next instruction.

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 content of memory location DS:000CH in the data window.

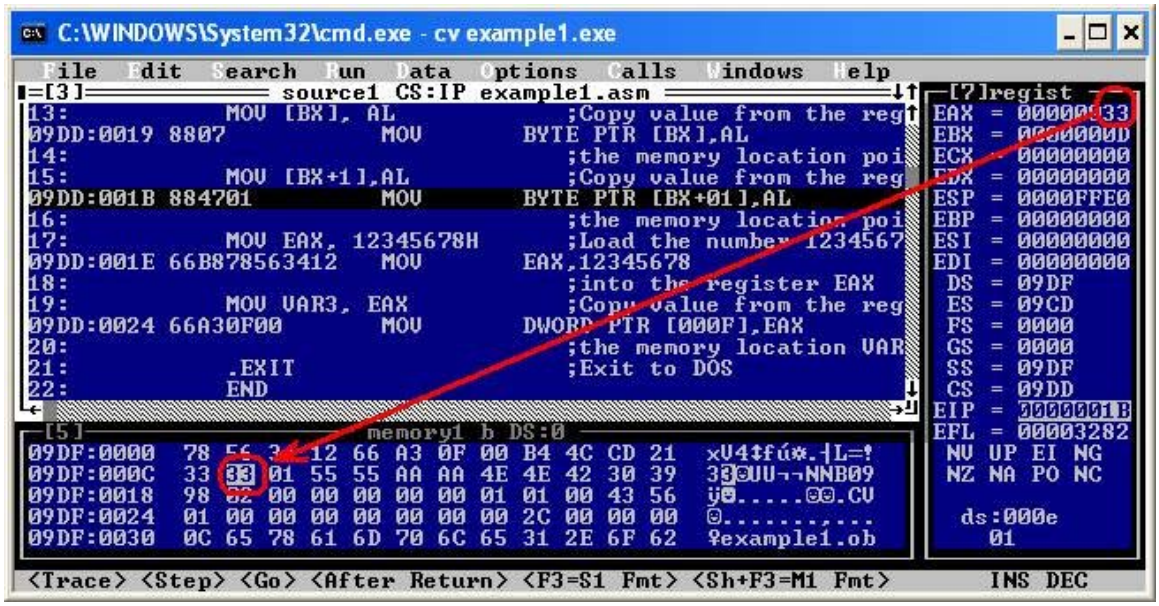
Now execute this instruction by pressing “F10”. Content of the register AL changed, taking the value from the VAR1.



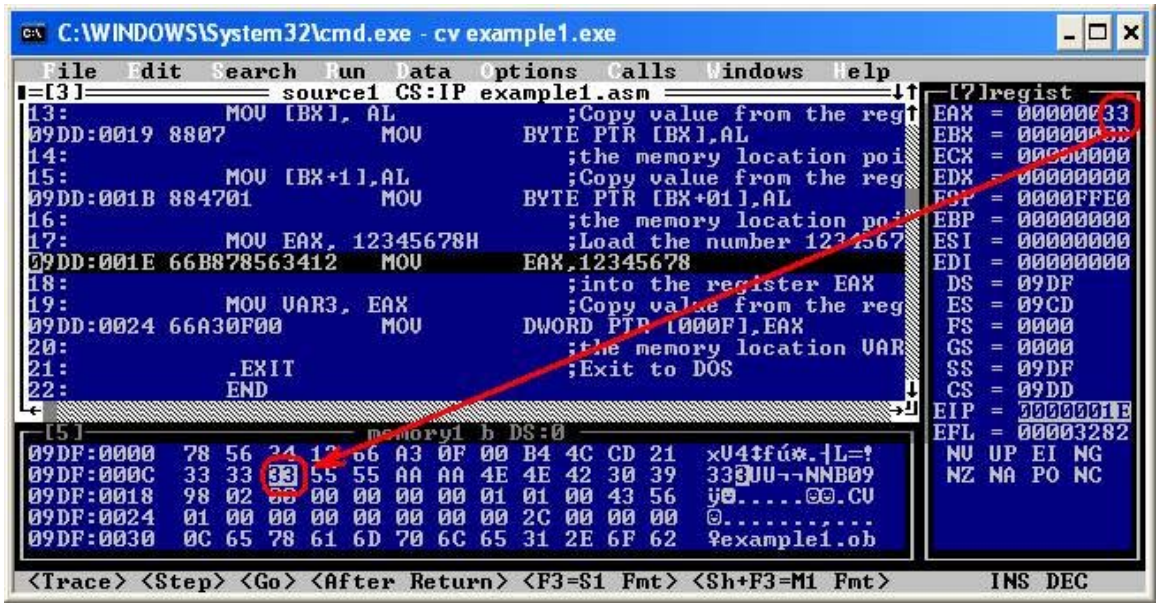
The next instruction is “MOV BX, OFFSET VAR2”. VAR2 follows VAR1 in memory and has offset of 000DH. This is the value that will be placed into the BX upon execution of this instruction. Press “F10” to execute.



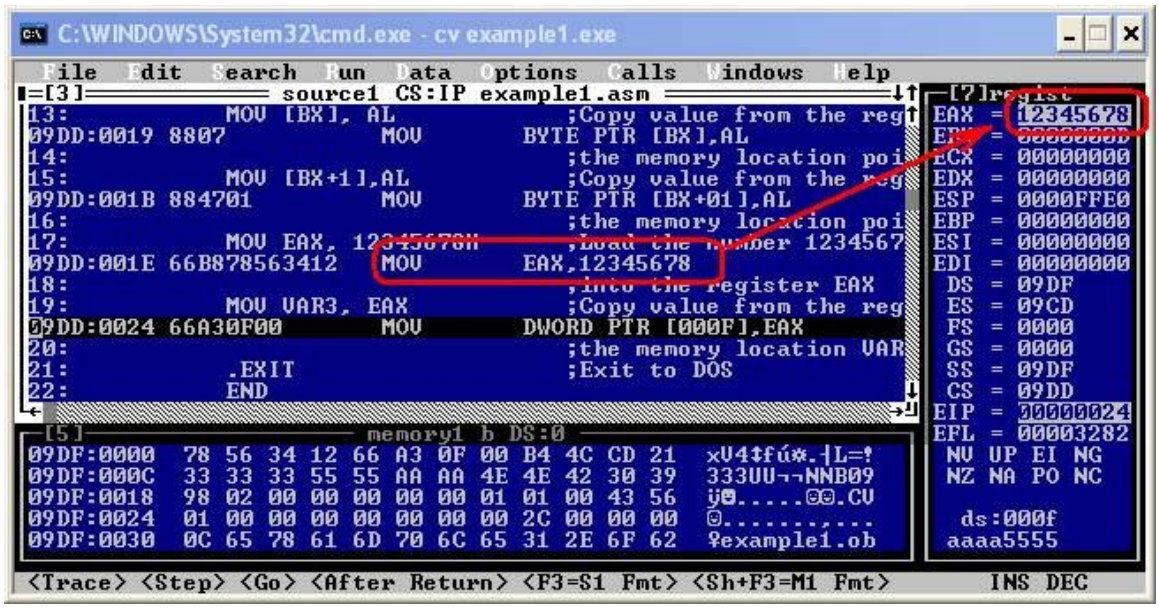
The following instruction “MOV [BX], AL” will copy the content of AL into the memory location pointed by BX within the data segment. After the previous instruction BX contains the offset of the first byte of VAR2 or 000DH. That is where the data from AL will appear. Press “F10” to execute. Note the debugger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



Instruction “MOV [BX+1], AL” will copy the content of the 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. That is the second byte of the VAR2. Press “F10” to execute. Note the change in the memory content.

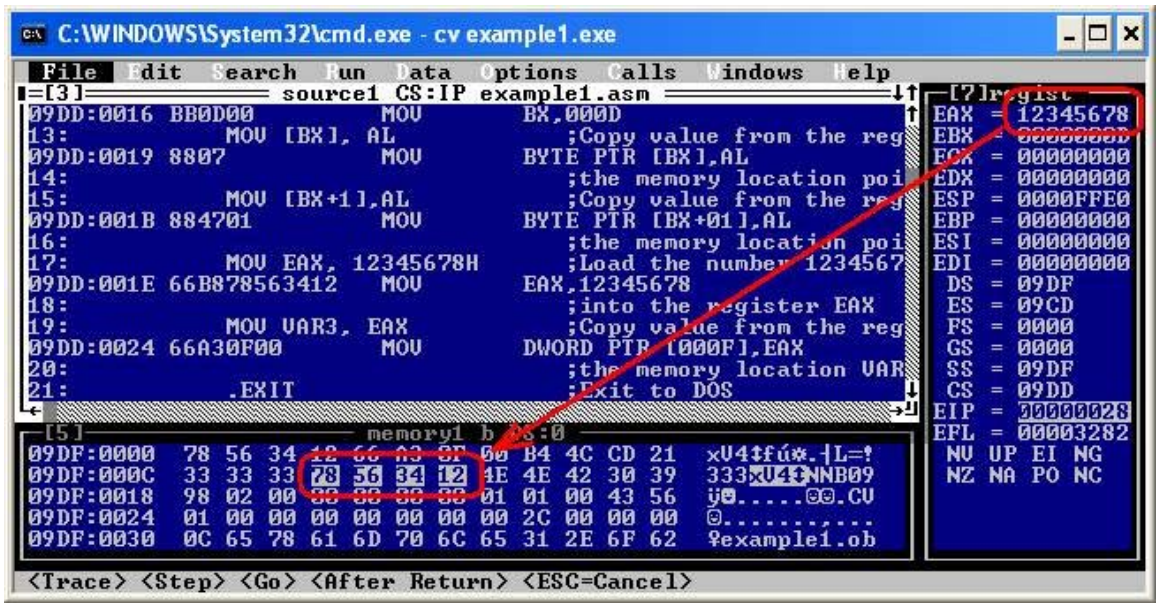


Instruction “MOV EAX, 12345678H” will place number 12345678H into the register EAX. Press “F10” to execute.

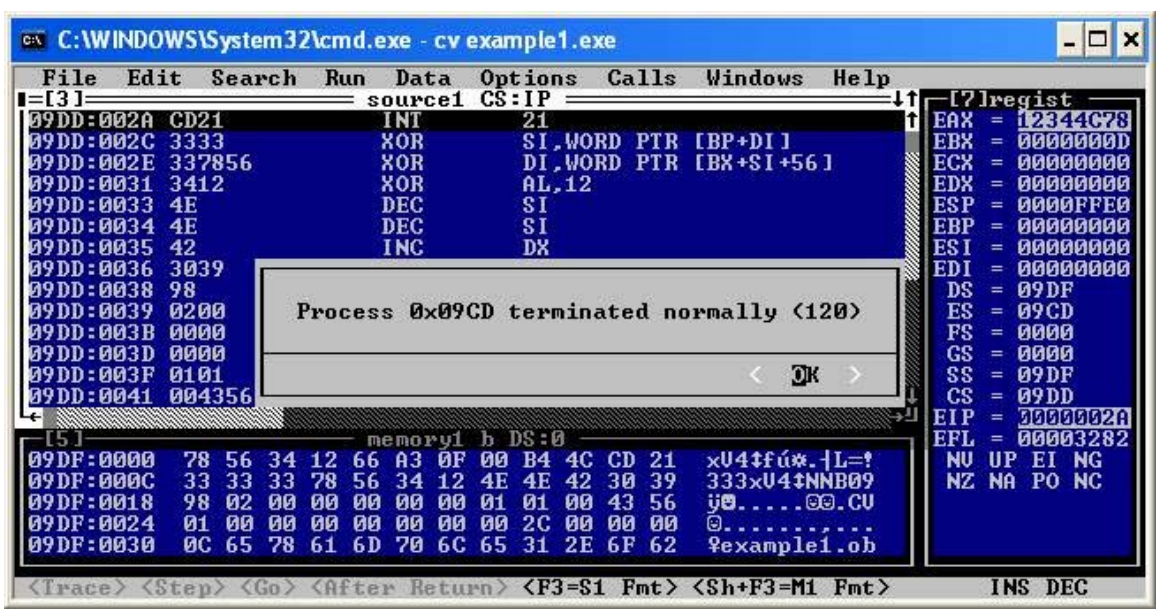


The instruction “MOV VAR3, EAX” became “MOV DWORD PTR [000F], EAX”.

VAR3 has been replaced by the actual offset (000FH) of VAR3 in the data memory. This instruction will take the content of the EAX and place into the four consecutive bytes of memory (a 32-bit variable) starting with the offset 000FH. Press “F10” to execute.



That 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” until the process terminates.



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