TOOLS - MASM VERSION

Important note before you start TOOLS.
You must have previously installed DOSBox and MASM to continue with TOOLS.
Instructions for installing DOSBOx and MASM are the class WEB site.

The purpose of TOOLS is to assure that you can access the assembler, linker, and debugger.

- Part A gets the sample files and verifies you can access the development tools.
- Part B gets you close to the hardware by using the debugger at a very basic level.
- Part C introduces three programs that will be of value as you develop code in this class.

Part A – Get the needed files from the Moodle WEB site

- Using a WEB browser, go the class Moodle WEB site.

  Go to the Moodle block titled *Homework Specifications HW0-HW8 And TOOLS*.

  Select the file hellom.asm and save it in your \P23X\HELLO directory.
  Select the file samples.exe and save it in your \P23X\SAMPLES directory.

  Warning ... be careful that your browser does not change the extension of the files

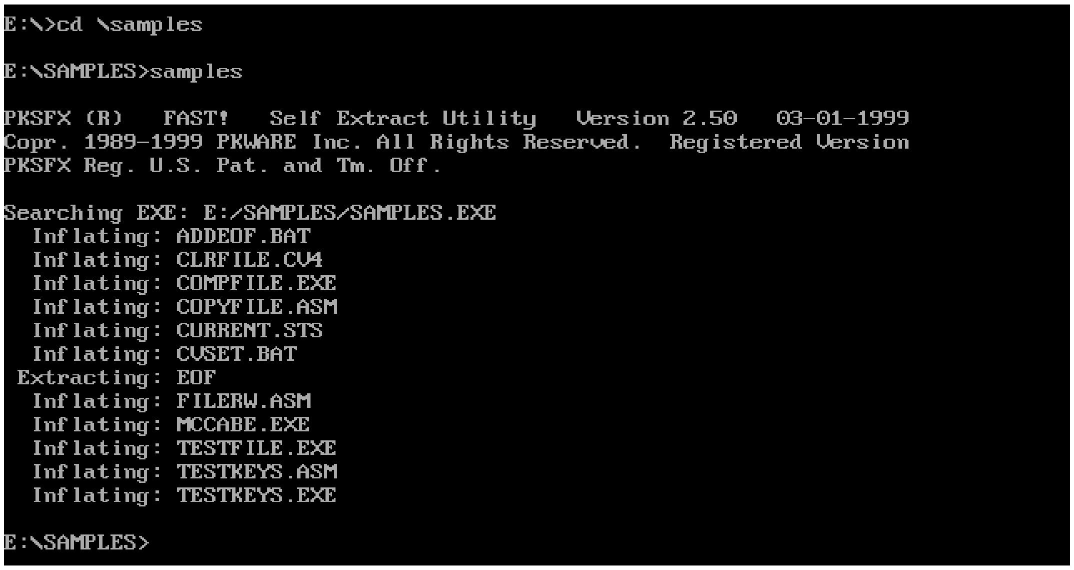
Start DOSBox

- Click on the DOSBox icon to open DOSBox
- Mount our subdirectories by typing: **mount e c:\P23X**
- Change to the p23x directory by typing: **e:**
- Set the path so DOSBox can find the assembler and linker by tying: **dbset**

Unpack the samples files

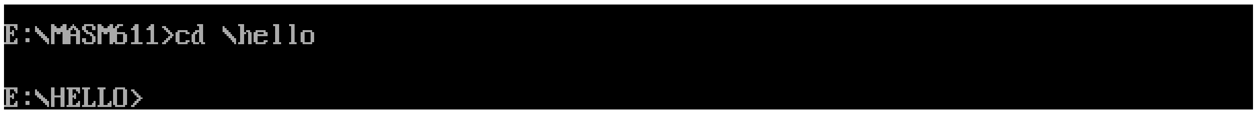
- Change to the samples directory by typing: **cd \samples**
- Type this command to unpack the samples: **samples**

In DOSBox you have two options for the window size: Standard small window Full screen
You switch between those two by hitting the *Alt + Enter* keys



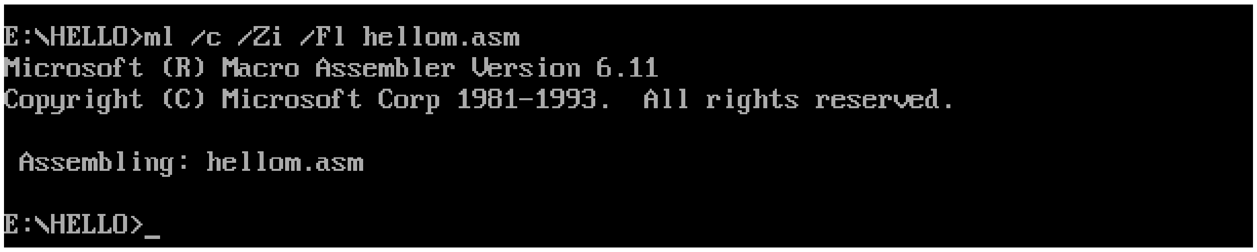
```
E:\>cd \samples

E:\SAMPLES>samples

PKSFX (R)   FAST!   Self Extract Utility   Version 2.50   03-01-1999
Copr. 1989-1999 PKWARE Inc. All Rights Reserved.  Registered Version
PKSFX Reg. U.S. Pat. and Tm. Off.

Searching EXE: E:/SAMPLES/SAMPLES.EXE
  Inflating: ADDEOF.BAT
  Inflating: CLRFILE.CV4
  Inflating: COMPFILE.EXE
  Inflating: COPYFILE.ASM
  Inflating: CURRENT.STS
  Inflating: CVSET.BAT
 Extracting: EOF
  Inflating: FILERW.ASM
  Inflating: MCCABE.EXE
  Inflating: TESTFILE.EXE
  Inflating: TESTKEYS.ASM
  Inflating: TESTKEYS.EXE

E:\SAMPLES>
```

Change to the hello directory with the hellom.asm source by typing: **cd \hello**


```
E:\MASM611>cd \hello

E:\HELLO>
```

Assemble the HELLOM program by typing: **ml /c /Zi /Fl hellom.asm**

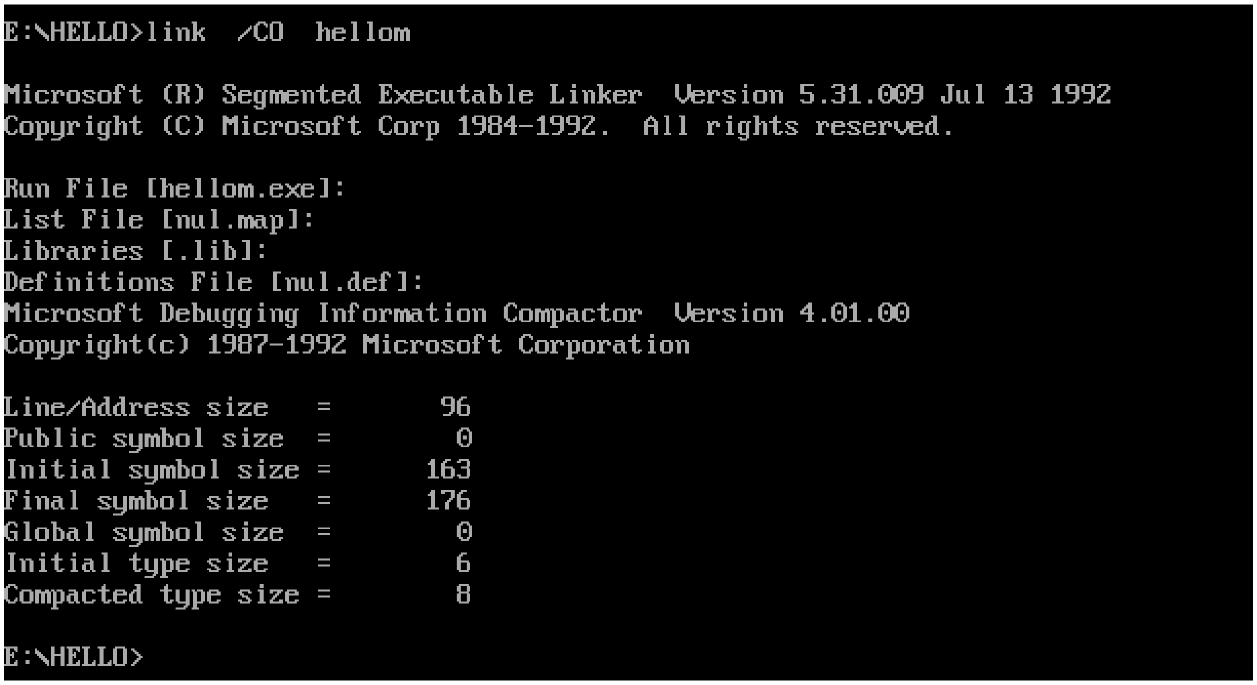
(Function of the parameters: /c compiles the code, /Zi includes debugger information, /Fl creates a listing file)


```
E:\HELLO>ml /c /Zi /Fl hellom.asm
Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993.  All rights reserved.

 Assembling: hellom.asm

E:\HELLO>_
```

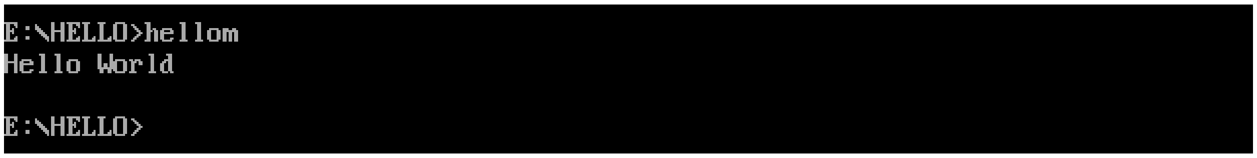
Link the HELLOM program by typing: **link /CO hellom**

(Function of the parameter: /CO includes debugger information)

You will be prompted for file names ... just press the enter key to take the default


```
E:\HELLO>link  /CO  hellom

Microsoft (R) Segmented Executable Linker  Version 5.31.009 Jul 13 1992
Copyright (C) Microsoft Corp 1984-1992.  All rights reserved.

Run File [hellom.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:
Microsoft Debugging Information Compactor  Version 4.01.00
Copyright(c) 1987-1992 Microsoft Corporation

Line/Address size   =       96
Public symbol size  =        0
Initial symbol size =      163
Final symbol size   =      176
Global symbol size  =        0
Initial type size   =        6
Compacted type size =        8

E:\HELLO>
```

Run the HELLOM program by typing: **hellom**


```
E:\HELLO>hellom
Hello World

E:\HELLO>
```

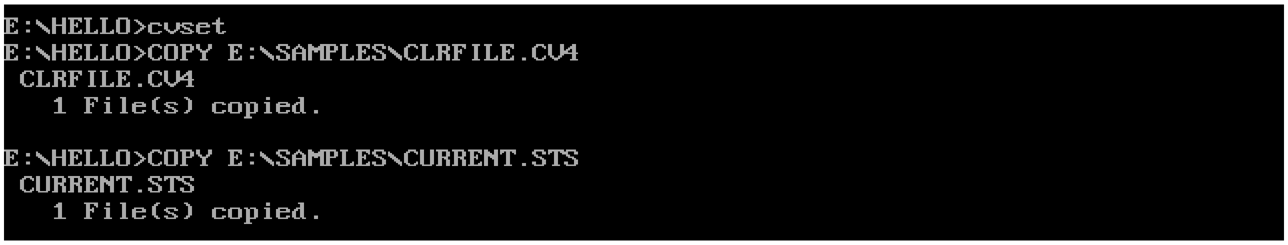
The tools are now be working and we can move to Part B to get a closer to the hardware.

Part B - Getting close to the hardware using the debugger

- On page 10 of this handout is a *CodeView Command Summary Under DOSBox.* Print and save it for future use.

  Also there is a copy of the file TOOLSSUB.TXT with the questions to be answered for this assignment.

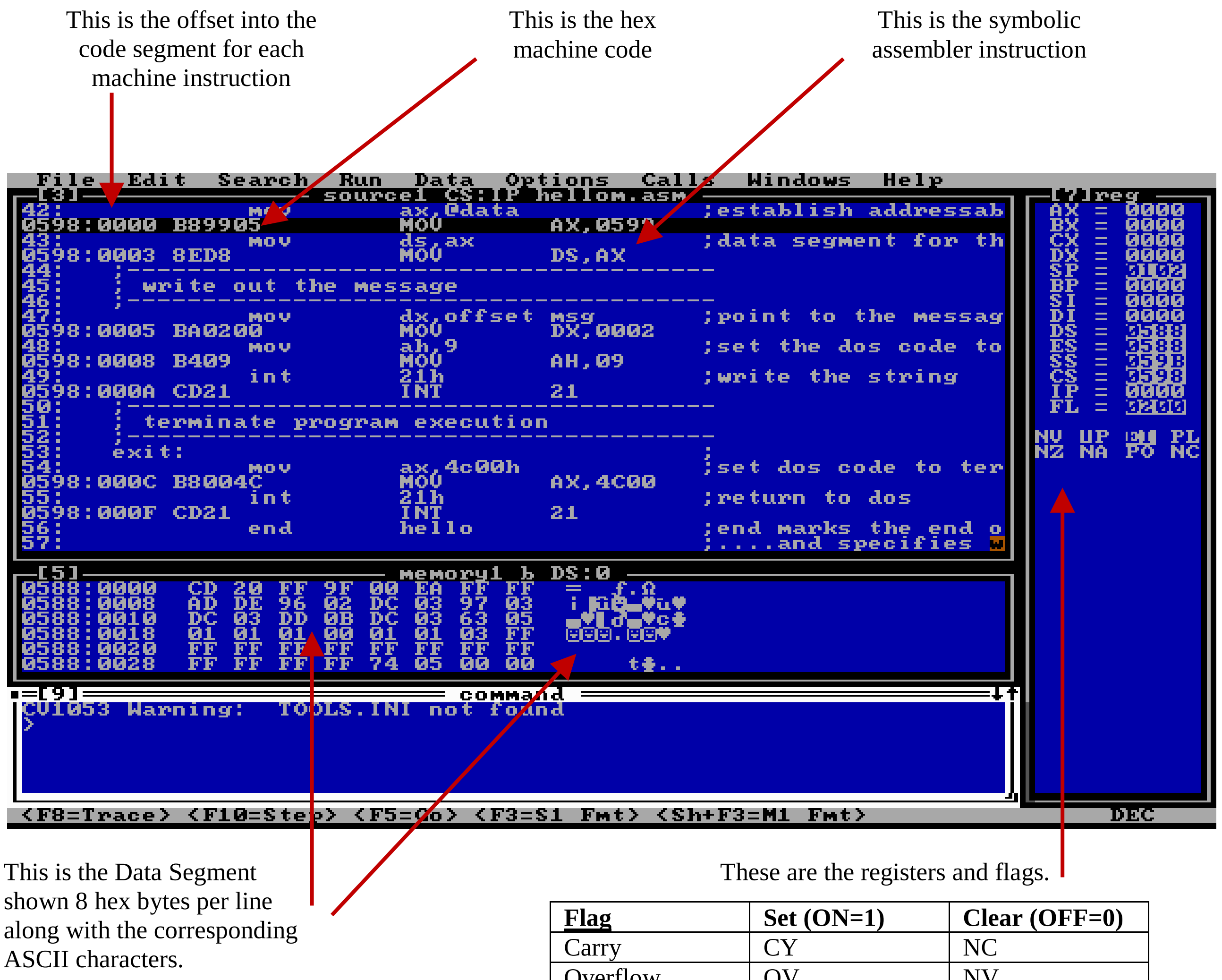
- Copy CodeView configuration files to your working directory (HELLO) . These files are in the SAMPLES directory. You can get them by typing: **`cvset`**



```
E:\HELLO>cvset
E:\HELLO>COPY E:\SAMPLES\CLRFILE.CV4
 CLRFILE.CV4
   1 File(s) copied.

E:\HELLO>COPY E:\SAMPLES\CURRENT.STS
 CURRENT.STS
   1 File(s) copied.
```

- Load the executable program into the debugger by typing: **`cv  hellom`**

- Below is the display for the CodeView debugger that you should see if the configuration files were correctly copied.



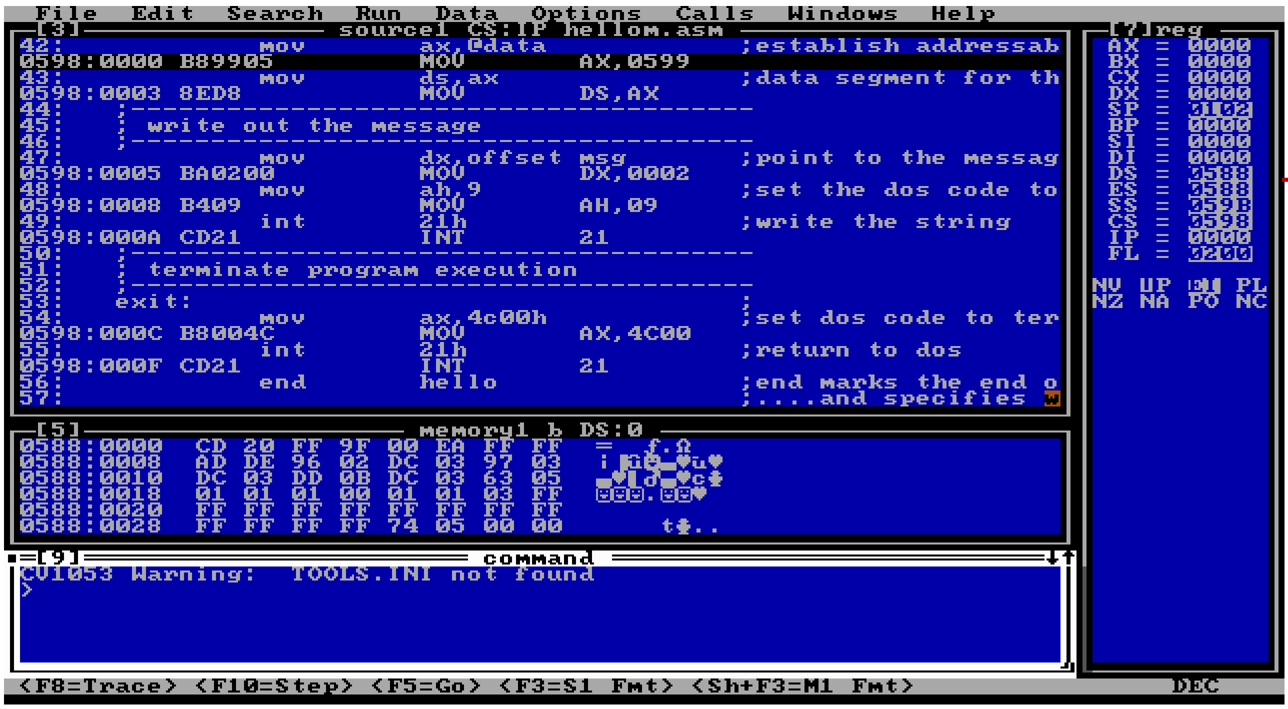
This is the offset into the code segment for each machine instruction
This is the hex machine code
This is the symbolic assembler instruction
This is the Data Segment shown 8 hex bytes per line along with the corresponding ASCII characters.
These are the registers and flags.

```
  File   Edit   Search   Run   Data   Options   Calls   Windows   Help
-[3]---------------- source1  CS:IP hellom.asm ----------- [7]reg -----
                mov          ax,@data        ;establish addressab   AX = 0000
0598:0000 B89905             MOV         AX,0599                    BX = 0000
                mov          ds,ax           ;data segment for th   CX = 0000
0598:0003 8ED8               MOV         DS,AX                      DX = 0000
          ;-------------------------------------------              SP = 0100
          ; write out the message                                   BP = 0000
          ;-------------------------------------------              SI = 0000
                mov          dx,offset msg   ;point to the messag   DI = 0000
0598:0005 BA0200             MOV         DX,0002                    DS = 0588
                mov          ah,9            ;set the dos code to   ES = 0588
0598:0008 B409               MOV         AH,09                      SS = 059B
                int          21h             ;write the string      CS = 0598
0598:000A CD21               INT         21                         IP = 0000
          ;-------------------------------------------              FL = 0200
          ; terminate program execution
          ;-------------------------------------------            NV UP EI PL
          exit:                              ;                    NZ NA PO NC
                mov          ax,4c00h        ;set dos code to ter
0598:000C B8004C             MOV         AX,4C00
                int          21h             ;return to dos
0598:000F CD21               INT         21
                end          hello           ;end marks the end o
                                             ;....and specifies
-[5]---------------- memory1 b DS:0 --------------------
0588:0000   CD 20 FF 9F 00 EA FF FF   =  ƒ.Ω
0588:0008   AD DE 96 02 DC 03 97 03   ¡
0588:0010   DC 03 DD 0B DC 03 63 05
0588:0018   01 01 01 00 01 01 03 FF   ☺☺☺.☺☺♥
0588:0020   FF FF FF FF FF FF FF FF
0588:0028   FF FF FF FF 74 05 00 00       t♣..
■=[9]================= command ========================↓↑
CV1053 Warning:  TOOLS.INI not found
>
 <F8=Trace> <F10=Step> <F5=Go> <F3=S1 Fmt> <Sh+F3=M1 Fmt>          DEC
```

Flag	**Set (ON=1)**	**Clear (OFF=0)**
Carry	CY	NC
Overflow	OV	NV
Sign	NG	PL
Zero	ZR	NZ

Later in the term you will learn how to create hex machine instructions from symbolic assembler code.

We can now trace the execution of your program. Answers to these questions are on the WEB.



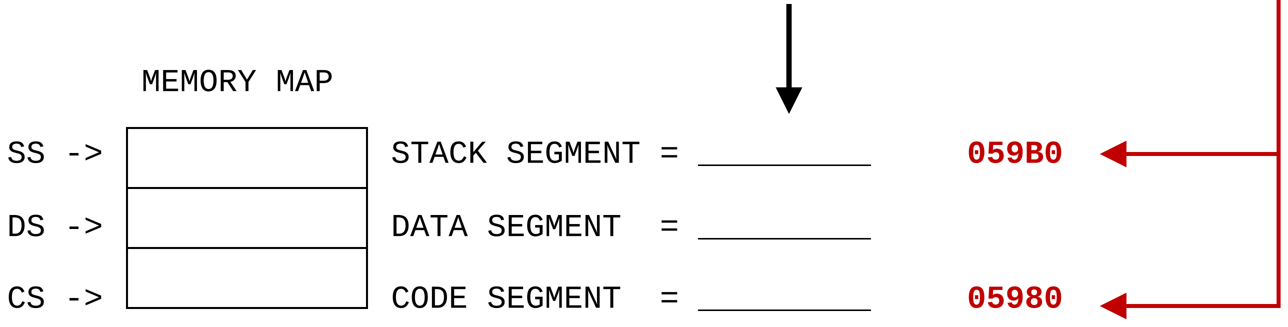
```
 File  Edit  Search  Run  Data  Options  Calls  Windows  Help
[3]            source1 CS:IP hellom.asm                        [7]reg
               mov        ax,@data        ;establish addressab  AX = 0000
0598:0000 B89905          MOV      AX,0599                       BX = 0000
               mov        ds,ax           ;data segment for th  CX = 0000
0598:0003 8ED8            MOV      DS,AX                         DX = 0000
        ;-----------------------------------                     SP = 0102
        ; write out the message                                  BP = 0000
        ;-----------------------------------                     SI = 0000
               mov        dx,offset msg   ;point to the messag  DI = 0000
0598:0005 BA0200          MOV      DX,0002                       DS = 0588
               mov        ah,9            ;set the dos code to  ES = 0588
0598:0008 B409            MOV      AH,09                         SS = 059B
               int        21h             ;write the string      CS = 0598
0598:000A CD21            INT      21                            IP = 0000
        ;-----------------------------------                     FL = 0200
        ; terminate program execution
        ;-----------------------------------                    NV UP EI PL
       exit:                              ;                     NZ NA PO NC
               mov        ax,4c00h        ;set dos code to ter
0598:000C B8004C          MOV      AX,4C00
               int        21h             ;return to dos
0598:000F CD21            INT      21
               end        hello           ;end marks the end o
                                          ;....and specifies
[5]            memory1 b DS:0
0588:0000  CD 20 FF 9F 00 EA FF FF   =  f.Ω
0588:0008  AD DE 96 02 DC 03 97 03
0588:0010  DC 03 DD 0B DC 03 63 05
0588:0018  01 01 01 00 01 01 03 FF
0588:0020  FF FF FF FF FF FF FF FF
0588:0028  FF FF FF FF 74 05 00 00       t..
[9]            command
CV1053 Warning:  TOOLS.INI not found
>
<F8=Trace> <F10=Step> <F5=Go> <F3=S1 Fmt> <Sh+F3=M1 Fmt>        DEC
```

Remember that the values in the Segment Registers (DS, ES, SS, CS) may be different for you.
The exact values are dependent on your environment when you run the program.

1. DOS has already setup the code segment (CS) and stack segment (SS) registers for you.

   Fill in your system's **absolute values** of the **code segment and the stack segment** on the picture below.

   The absolute value of the segment is the value in the segment register with a 0 added to the end.



The data segment register (DS) and extra segment (ES) register will be pointing to a DOS control block, **not** to your data.

We cannot yet fill in the absolute value for data segment since our program has not initialized DS register yet.

The function of the first two instructions is to initialize the data segment register.

PS ... this is why you do not see the “Hello World” message in the data segment (memory1 window).

2. The first instruction to be executed is: **mov ax, @data**.

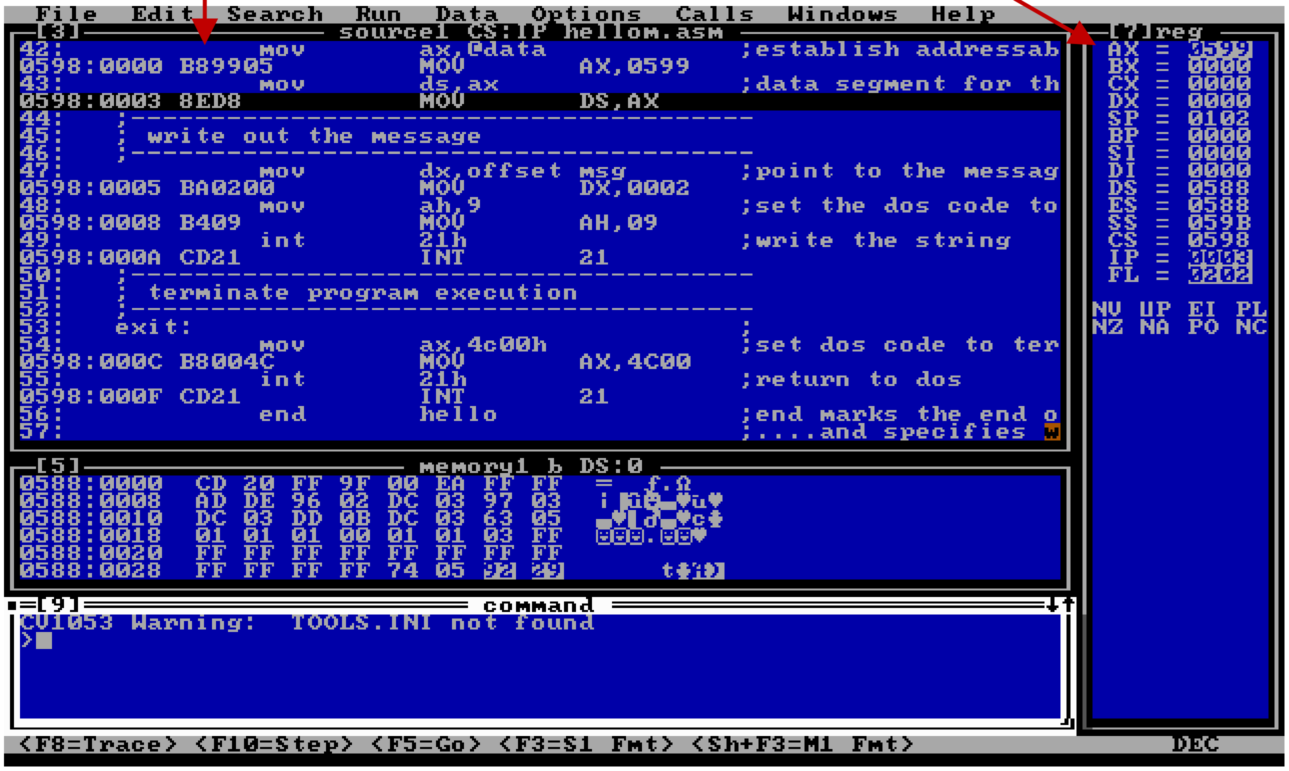
   It is located as cs:0 which is zero bytes into the code segment.

   The machine code is: B8 ________ ________ (fill in the blanks for your system).

   The B8 is the operation code which tells the hardware to load the ax register with the two hex bytes that follow (99 05).

   Press F8 to execute the instruction.

After executing the instruction the value in the ax register is reversed from the machine code. This is due to byte swapping where words are stored backwards in memory . So for this system the machine code is B8 99 05 and the value loaded into the ax register is 05 99.

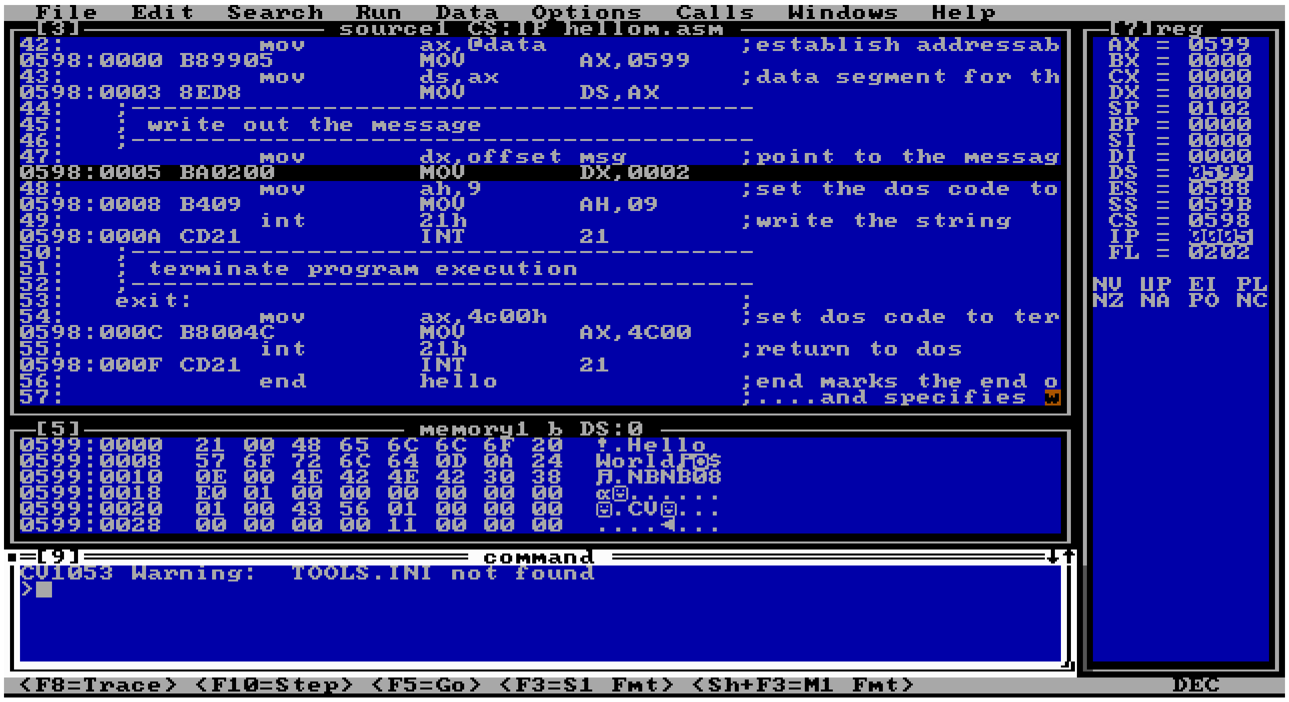


```
 File  Edit  Search  Run  Data  Options  Calls  Windows  Help
[3]            source1 CS:IP hellom.asm                          [7]reg
                     mov       ax,@data           ;establish addressab   AX = 0599
0598:0000 B89905               MOV       AX,0599                         BX = 0000
                     mov       ds,ax              ;data segment for th   CX = 0000
0598:0003 8ED8                 MOV       DS,AX                           DX = 0000
              ;----------------------------------                         SP = 0102
              ; write out the message                                    BP = 0000
              ;----------------------------------                         SI = 0000
                     mov       dx,offset msg      ;point to the messag   DI = 0000
0598:0005 BA0200               MOV       DX,0002                         DS = 0588
                     mov       ah,9               ;set the dos code to   ES = 0588
0598:0008 B409                 MOV       AH,09                           SS = 059B
                     int       21h                ;write the string      CS = 0598
0598:000A CD21                 INT       21                              IP = 0003
              ;----------------------------------                         FL = 3202
              ; terminate program execution
              ;----------------------------------                        NV UP EI PL
              exit:                               ;                      NZ NA PO NC
                     mov       ax,4c00h           ;set dos code to ter
0598:000C B8004C               MOV       AX,4C00
                     int       21h                ;return to dos
0598:000F CD21                 INT       21
                     end       hello              ;end marks the end o
                                                  ;....and specifies
[5]            memory1 b DS:0
0588:0000  CD 20 FF 9F 00 EA FF FF
0588:0008  AD DE 96 02 DC 03 97 03
0588:0010  DC 03 DD 0B DC 03 63 05
0588:0018  01 01 01 00 01 01 03 FF
0588:0020  FF FF FF FF FF FF FF FF
0588:0028  FF FF FF FF 74 05 92 29
[9]            command
CV1053 Warning:  TOOLS.INI not found
>
<F8=Trace> <F10=Step> <F5=Go> <F3=S1 Fmt> <Sh+F3=M1 Fmt>            DEC
```

3. The next instruction to be executed is: **mov ds,ax**.   It loads the ds register with the correct value.

   Press F8 to execute the instruction.

   Now that we have initialized the ds register we can add the absolute value of data segment to the memory map below.



```
 File   Edit   Search   Run   Data   Options   Calls   Windows   Help
-[3]------------------ source1 CS:IP hellom.asm ---------------   -[7]reg
              mov            ax,@data              ;establish addressab   AX = 0599
0598:0000 B89905             MOV        AX,0599                          BX = 0000
              mov            ds,ax                 ;data segment for th   CX = 0000
0598:0003 8ED8               MOV        DS,AX                            DX = 0000
        ;---------------------------------------------                    SP = 0102
        ; write out the message                                           BP = 0000
        ;---------------------------------------------                    SI = 0000
              mov            dx,offset msg         ;point to the messag   DI = 0000
0598:0005 BA0200             MOV        DX,0002                          DS = 0599
              mov            ah,9                  ;set the dos code to   ES = 0588
0598:0008 B409               MOV        AH,09                            SS = 059B
              int            21h                   ;write the string      CS = 0598
0598:000A CD21               INT        21                               IP = 0005
        ;---------------------------------------------                    FL = 0202
        ; terminate program execution
        ;---------------------------------------------                    NV UP EI PL
        exit:                                      ;                      NZ NA PO NC
              mov            ax,4c00h              ;set dos code to ter
0598:000C B8004C             MOV        AX,4C00
              int            21h                   ;return to dos
0598:000F CD21               INT        21
              end            hello                 ;end marks the end o
                                                   ;....and specifies
-[5]----------------- memory1 b DS:0 ---------------------
0599:0000  21 00 48 65 6C 6C 6F 20  !.Hello
0599:0008  57 6F 72 6C 64 0D 0A 24  World♪◙$
0599:0010  0E 00 4E 42 4E 42 30 38  ♫.NBNB08
0599:0018  E0 01 00 00 00 00 00 00  α☺......
0599:0020  01 00 43 56 01 00 00 00  ☺.CV☺...
0599:0028  00 00 00 00 11 00 00 00  ....◄...
■=[9]==================== command ========================↓↑
CV1053 Warning:  TOOLS.INI not found
>
 <F8=Trace> <F10=Step> <F5=Go> <F3=S1 Fmt> <Sh+F3=M1 Fmt>                  DEC
```

MEMORY MAP

SS ->		STACK SEGMENT =	________	059B0
DS ->		DATA SEGMENT =	________	05990
CS ->		CODE SEGMENT =	________	05980

Note that now that the DS register has been initialized, you can see the "Hello World" message in the data segment.

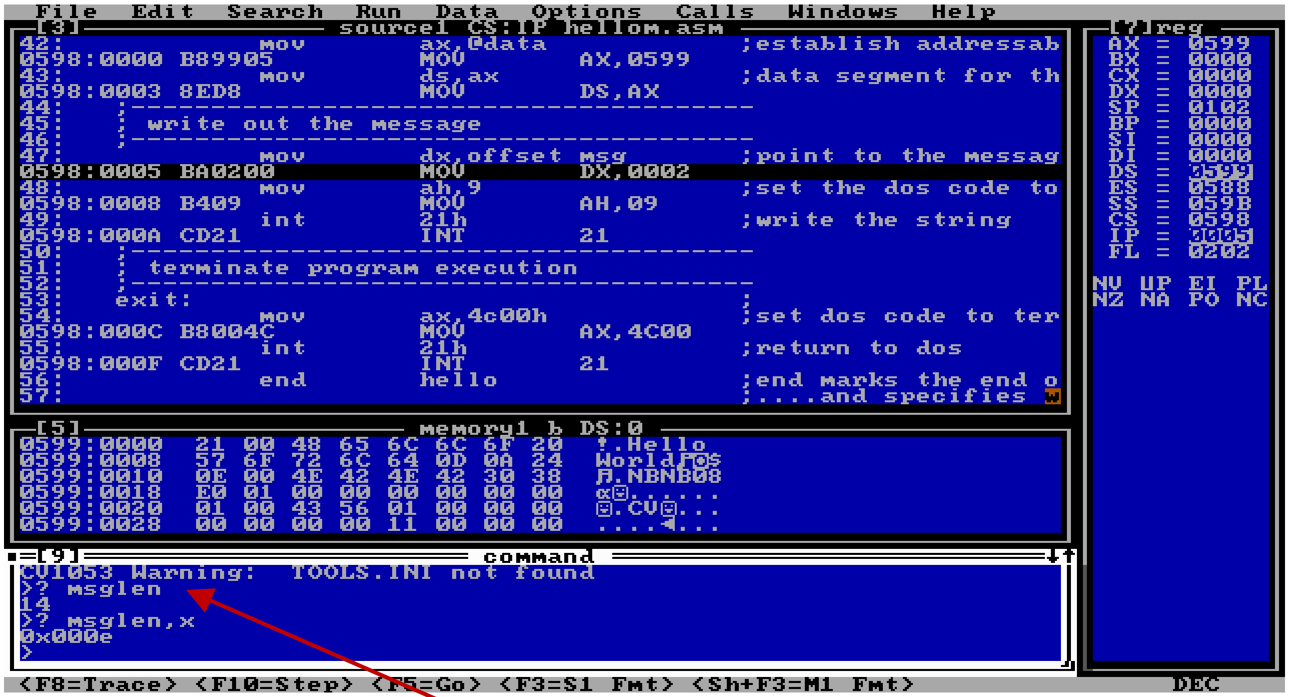
Stop.

You will need these three values as answers to the questions in toolssub.txt

The bottom most window is the Command Window. Make the Command Window [9] the active window by clicking the mouse in the Command Window.

Let’s display a variable. The word size variable *msglen* contains the number of characters in the “Hello World” message.

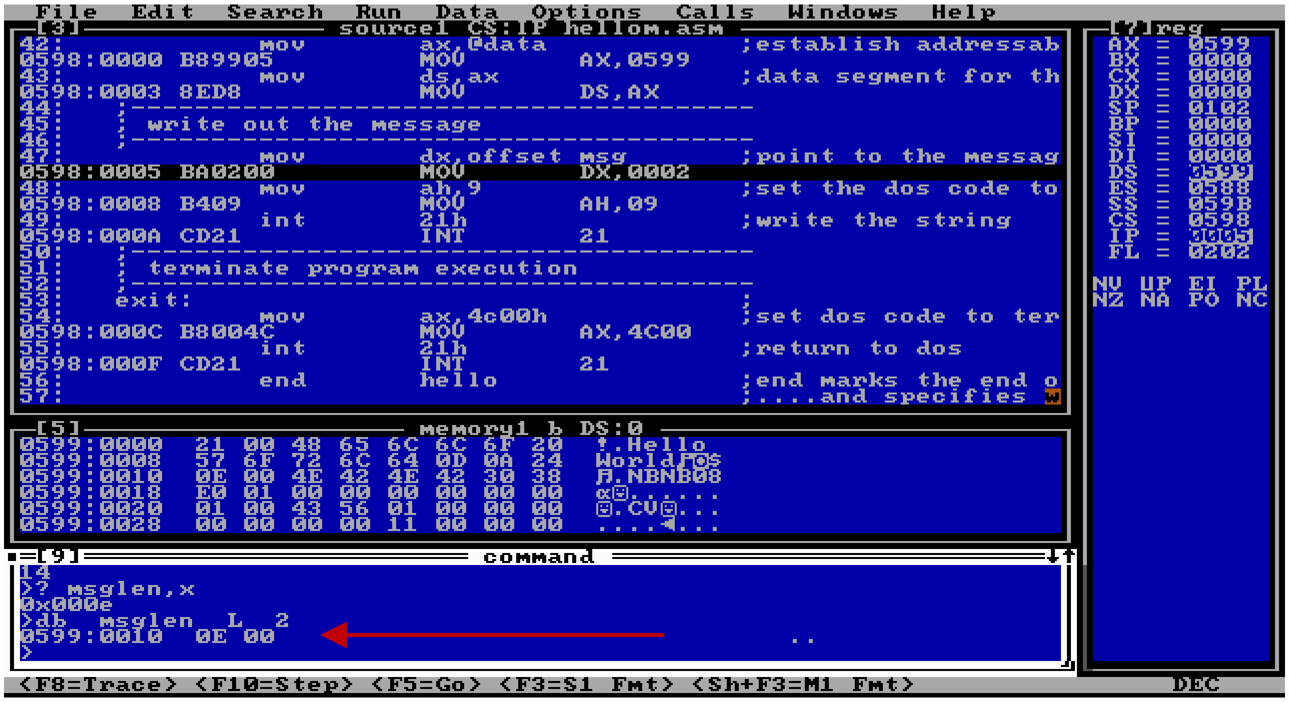
That count is 14 (Hello World = 11, Carriage Return and Line Feed = 2, DOS string terminator $ = 1)


File Edit Search Run Data Options Calls Windows Help
<F8=Trace> <F10=Step> <F5=Go> <F3=S1 Fmt> <Sh+F3=M1 Fmt>
DEC

```
[3] source1 CS:IP hellom.asm
                 mov          ax,@data            ;establish addressab
0598:0000 B89905                 MOV          AX,0599
                 mov          ds,ax               ;data segment for th
0598:0003 8ED8                   MOV          DS,AX
       ;------------------------------------------------
       ; write out the message
       ;------------------------------------------------
                 mov          dx,offset msg       ;point to the messag
0598:0005 BA0200                 MOV          DX,0002
                 mov          ah,9                ;set the dos code to
0598:0008 B409                   MOV          AH,09
                 int          21h                 ;write the string
0598:000A CD21                   INT          21
       ;------------------------------------------------
       ; terminate program execution
       ;------------------------------------------------
       exit:                                      ;
                 mov          ax,4c00h            ;set dos code to ter
0598:000C B8004C                 MOV          AX,4C00
                 int          21h                 ;return to dos
0598:000F CD21                   INT          21
                 end          hello               ;end marks the end o
                                                  ;....and specifies

[7] reg
AX = 0599
BX = 0000
CX = 0000
DX = 0000
SP = 0102
BP = 0000
SI = 0000
DI = 0000
DS = 0599
ES = 0588
SS = 059B
CS = 0598
IP = 0005
FL = 0202

NV UP EI PL
NZ NA PO NC

[5] memory1 b DS:0
0599:0000  21 00 48 65 6C 6C 6F 20  !.Hello
0599:0008  57 6F 72 6C 64 0D 0A 24  World♪◙$
0599:0010  0E 00 4E 42 4E 42 30 38  ♫.NBNB08
0599:0018  E0 01 00 00 00 00 00 00  α☺......
0599:0020  01 00 43 56 01 00 00 00  ☺.CV☺...
0599:0028  00 00 00 00 11 00 00 00  ....◄...

[9] command
CV1053 Warning: TOOLS.INI not found
>? msglen
14
>? msglen,x
0x000e
>
```

In the command window type: **? msglen** It will display the 14.

In the command window type: **? msglen,x** It will display the hex value of 14 = 000E

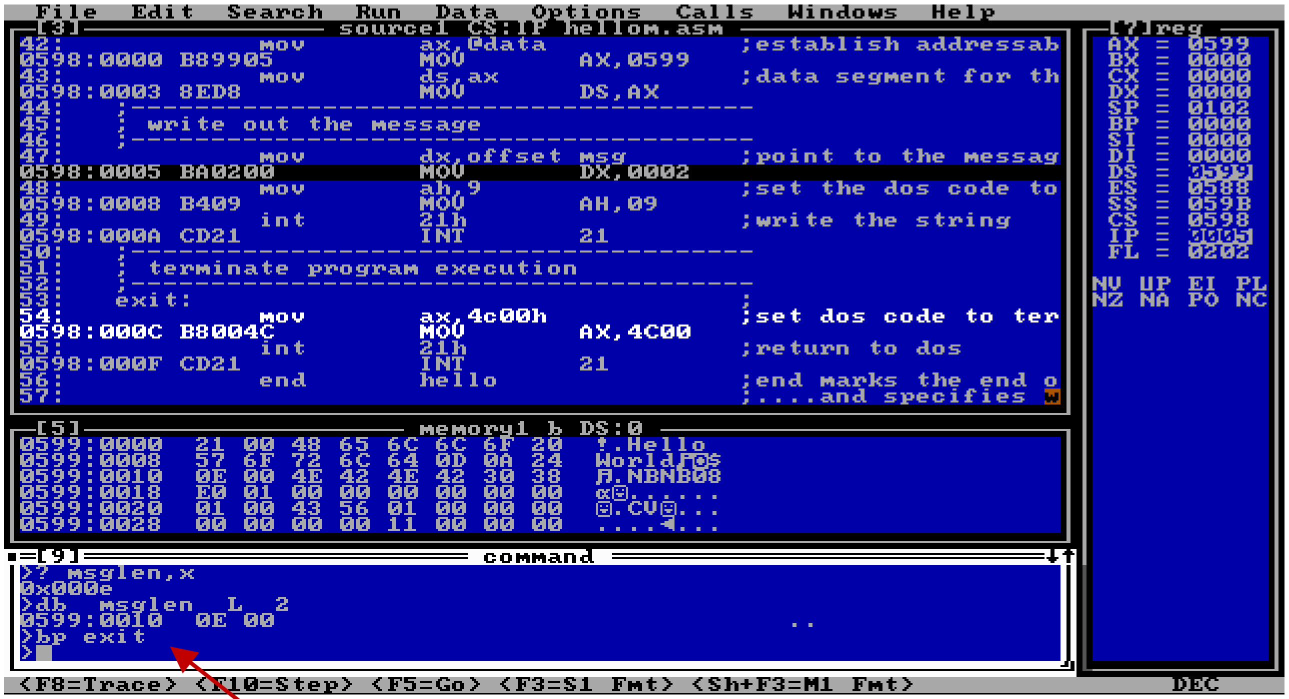

File Edit Search Run Data Options Calls Windows Help
<F8=Trace> <F10=Step> <F5=Go> <F3=S1 Fmt> <Sh+F3=M1 Fmt>
DEC

```
[3] source1 CS:IP hellom.asm
                 mov          ax,@data            ;establish addressab
0598:0000 B89905                 MOV          AX,0599
                 mov          ds,ax               ;data segment for th
0598:0003 8ED8                   MOV          DS,AX
       ;------------------------------------------------
       ; write out the message
       ;------------------------------------------------
                 mov          dx,offset msg       ;point to the messag
0598:0005 BA0200                 MOV          DX,0002
                 mov          ah,9                ;set the dos code to
0598:0008 B409                   MOV          AH,09
                 int          21h                 ;write the string
0598:000A CD21                   INT          21
       ;------------------------------------------------
       ; terminate program execution
       ;------------------------------------------------
       exit:                                      ;
                 mov          ax,4c00h            ;set dos code to ter
0598:000C B8004C                 MOV          AX,4C00
                 int          21h                 ;return to dos
0598:000F CD21                   INT          21
                 end          hello               ;end marks the end o
                                                  ;....and specifies

[7] reg
AX = 0599
BX = 0000
CX = 0000
DX = 0000
SP = 0102
BP = 0000
SI = 0000
DI = 0000
DS = 0599
ES = 0588
SS = 059B
CS = 0598
IP = 0005
FL = 0202

NV UP EI PL
NZ NA PO NC

[5] memory1 b DS:0
0599:0000  21 00 48 65 6C 6C 6F 20  !.Hello
0599:0008  57 6F 72 6C 64 0D 0A 24  World♪◙$
0599:0010  0E 00 4E 42 4E 42 30 38  ♫.NBNB08
0599:0018  E0 01 00 00 00 00 00 00  α☺......
0599:0020  01 00 43 56 01 00 00 00  ☺.CV☺...
0599:0028  00 00 00 00 11 00 00 00  ....◄...

[9] command
14
>? msglen,x
0x000e
>db msglen L 2
0599:0010  0E 00                                          ..
>
```

However, remember that msglen is really stored backwards in memory as 0E 00 in hex due to byte swapping.

You can see that by typing: **db msglen L 2** which displays the 2 bytes that make up msglen.

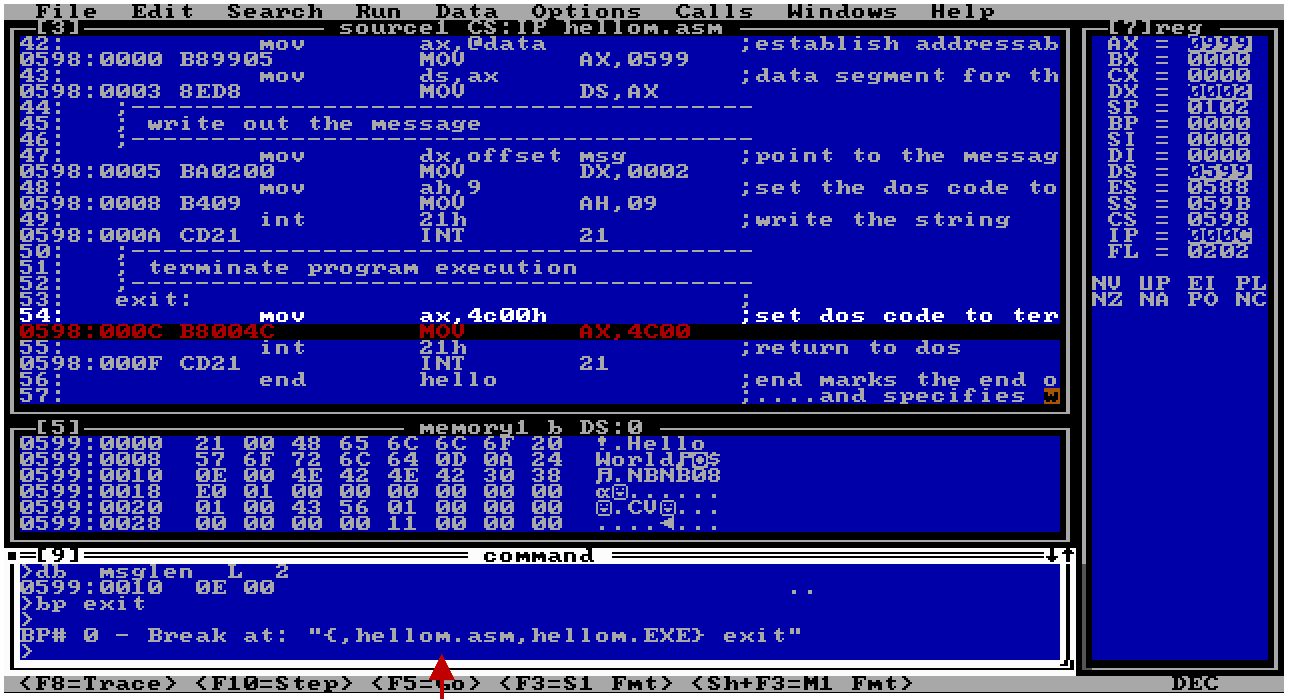
4. There will be times when you want to watch the execution of a specific part of your code. You want to skip over other code and get to this specific code. That is the function of the *breakpoint*. A breakpoint allows you to skip over code and stop at a specifically labeled instruction. To use a breakpoint you need to know the symbolic label name of the instruction that you want to go to. In the hellom program we have a label *exit*. Let's go to that label.



```
 File  Edit  Search  Run  Data  Options  Calls  Windows  Help
-[3]------------- source1 CS:IP hellom.asm -----------------   -[7]reg --
            mov         ax,@data             ;establish addressab   AX = 0599
0598:0000 B89905        MOV         AX,0599                         BX = 0000
            mov         ds,ax                ;data segment for th   CX = 0000
0598:0003 8ED8          MOV         DS,AX                           DX = 0000
      ;-------------------------------------------                  SP = 0102
      ; write out the message                                       BP = 0000
      ;-------------------------------------------                  SI = 0000
            mov         dx,offset msg        ;point to the messag   DI = 0000
0598:0005 BA0200        MOV         DX,0002                         DS = 0599
            mov         ah,9                 ;set the dos code to   ES = 0588
0598:0008 B409          MOV         AH,09                           SS = 059B
            int         21h                  ;write the string      CS = 0598
0598:000A CD21          INT         21                              IP = 0005
      ;-------------------------------------------                  FL = 0202
      ; terminate program execution
      ;-------------------------------------------                  NV UP EI PL
      exit:                                  ;                      NZ NA PO NC
            mov         ax,4c00h             ;set dos code to ter
0598:000C B8004C        MOV         AX,4C00
            int         21h                  ;return to dos
0598:000F CD21          INT         21
            end         hello                ;end marks the end o
                                             ;....and specifies
-[5]------------- memory1 b DS:0 -----------
0599:0000  21 00 48 65 6C 6C 6F 20  !.Hello
0599:0008  57 6F 72 6C 64 0D 0A 24  World♪◙$
0599:0010  0E 00 4E 42 4E 42 30 38  ♫.NBNB08
0599:0018  E0 01 00 00 00 00 00 00  α☺......
0599:0020  01 00 43 56 01 00 00 00  ☺.CV☺...
0599:0028  00 00 00 00 11 00 00 00  ....◄...
-[9]---------------- command ----------------
>? msglen,x
0x000e
>db  msglen  L  2
0599:0010  0E 00                                    ..
>bp exit
>
 <F8=Trace> <F10=Step> <F5=Go> <F3=S1 Fmt> <Sh+F3=M1 Fmt>        DEC
```

Type: **bp exit** to set a breakpoint at exit:

Now pres F5 to have the program run to that breakpoint.



```
 File  Edit  Search  Run  Data  Options  Calls  Windows  Help
-[3]------------- source1 CS:IP hellom.asm -----------------   -[7]reg --
            mov         ax,@data             ;establish addressab   AX = 0599
0598:0000 B89905        MOV         AX,0599                         BX = 0000
            mov         ds,ax                ;data segment for th   CX = 0000
0598:0003 8ED8          MOV         DS,AX                           DX = 0002
      ;-------------------------------------------                  SP = 0102
      ; write out the message                                       BP = 0000
      ;-------------------------------------------                  SI = 0000
            mov         dx,offset msg        ;point to the messag   DI = 0000
0598:0005 BA0200        MOV         DX,0002                         DS = 0599
            mov         ah,9                 ;set the dos code to   ES = 0588
0598:0008 B409          MOV         AH,09                           SS = 059B
            int         21h                  ;write the string      CS = 0598
0598:000A CD21          INT         21                              IP = 000C
      ;-------------------------------------------                  FL = 0202
      ; terminate program execution
      ;-------------------------------------------                  NV UP EI PL
      exit:                                  ;                      NZ NA PO NC
            mov         ax,4c00h             ;set dos code to ter
0598:000C B8004C        MOV         AX,4C00
            int         21h                  ;return to dos
0598:000F CD21          INT         21
            end         hello                ;end marks the end o
                                             ;....and specifies
-[5]------------- memory1 b DS:0 -----------
0599:0000  21 00 48 65 6C 6C 6F 20  !.Hello
0599:0008  57 6F 72 6C 64 0D 0A 24  World♪◙$
0599:0010  0E 00 4E 42 4E 42 30 38  ♫.NBNB08
0599:0018  E0 01 00 00 00 00 00 00  α☺......
0599:0020  01 00 43 56 01 00 00 00  ☺.CV☺...
0599:0028  00 00 00 00 11 00 00 00  ....◄...
-[9]---------------- command ----------------
>db  msglen  L  2
0599:0010  0E 00                                    ..
>bp exit

BP# 0 - Break at: "{,hellom.asm,hellom.EXE} exit"
>
 <F8=Trace> <F10=Step> <F5=Go> <F3=S1 Fmt> <Sh+F3=M1 Fmt>        DEC
```

The code now runs until it reaches exit.

Press F5 to execute the rest of the program. You should see a message that the program terminated.

Click the mouse on FILE at the top of the screen and then EXIT to take you back to the DOS window and prompt.

Part C - Utility programs (three tools of value when you start programming ... testfile compfile testkeys)

Your mouse is attached to the DOSBox window. To free the mouse press Ctrl-F10 or Alt-Tab

- Using any text editor, create in the HELLO directory a file named *test1.txt* with these two lines.

```
ABC
DEF
```

In DOSBox at the DOS prompt, type: **dir**

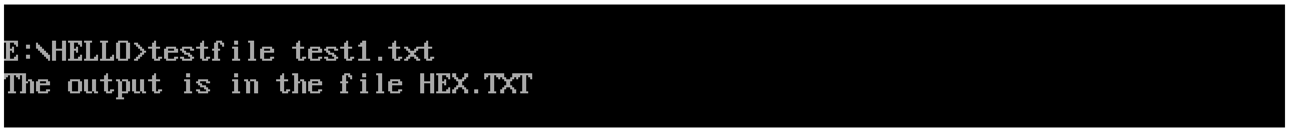
Did the file *test1.txt* appear? It may not. Files created outside of DOSBox are only discovered when you start DOSBox.

If the file does not appear, type *rescan* or press Ctrl-F4. Either should make the new file visible.

At the DOS prompt, type: **dir** Now the file *test1.txt* should now appear.

- Let us look at the hex contents of *test1.txt.*

Type **testfile test1.txt** which converts the contents to *test1.txt* to hex and writes it to the file *hex.txt.*



```
E:\HELLO>testfile test1.txt
The output is in the file HEX.TXT
```

Use your editor to look at *hex.txt.* I used Windows NOTEPAD to create test1.txt and this is what NOTEPAD created.

```
For this ASCII text:          A  B  C CR LF  D  E  F
Notepad created this hex data: 41 42 43 0D 0A 44 45 46
```

NOTEPAD did not put a CR/LF pair at the end of the last line because I did not hit *Enter* at the end of the last line when creating the file. It also did not put a DOS end-of-file character (1Ah) as the last character of the file? Many editors do not.

You can use this same procedure to see what your editor does.

- The other utility program is *compfile* which will show the hex vales in two files and compare the two files. To use, type

compfile file1 file2 The hex values in bot files will be placed in the file hex.txt

- The *testkeys* program will read any key you press and output the hex value of that key.

To run it, type: **testkeys**

Press some upper and lower case letters and numbers to see the hex ASCII values for those keys. Compare the hex values generated to those in the ASCII table in section 26 of the Class Notes.

Press the *Enter* key ... what hex value is generated?

Press the F1 function key. It generates 2 bytes.
Why? Read *Extended ASCII characters* in the File I/O section of the notes.

You will need these 2 values as answers to the questions in toolssub.txt

The program will terminate when you enter a period.

Answer the questions in the file TOOLSSUB.TXT.
Submit that file to the TOOLS submit locker and you are done.

CodeView Command Summary Under DOSBox

- Assure you have assembled and linked with the CodeView options
  ml /c /Zi /Fl *program.asm*
  link /CO *program*

- Assure you copied the CodeView configuration files into your working directory
  cvset

  Start CodeView by typing: **cv *program***

- Click the mouse in a window to make it the active window.
  You can also switch to a window, to make a hidden widow visible using the Alt Key
  (Alt+2 shows Watch window, Alt+3 shows Source window, Alt+5 shows the Memory window,
  Alt+7 shows the Register window, Alt+9 shows the Command Window)

- Common Function Keys

F5	Run the program at full speed until either: - the program ends - the program reaches a *breakpoint*
F8	Execute the next instruction. Also called *Single Step*.
F10	Execute the next instruction ... with a difference from F8. If it is a call to subroutine, execute the whole subroutine as if it were a single instruction. If the instruction is the loop instruction then repeatedly execute the loop until cx=0.

- Common Commands for the command window

? var	displays the variable as it was declared
? var,c	displays the variable as an ASCII character
? var,i	displays the variable as a signed integer
? var,u	displays the variable as an unsigned number
? var,x	displays the variable in hex
db start L #bytes	dumps hex bytes of memory starting at the start location (*db var L 8* dumps 8 hex bytes starting at var)
dw start L #bytes	dumps hex words of memory starting at the start location (*dw var L 8* dumps 8 hex words starting at var)
da start L #bytes	dumps ASCII characters of memory starting at the start location (*da var L 8* dumps 8 ASCII characters starting at var)
bp label	Set a breakpoint at the specified instruction
w? var w? by si w? wo si+12 w? by si+12,f	Watch the variable var Watch the byte size location pointed to by si. Watch the word size location pointed to by si+12. Watch the byte size location pointed to by si using the format defined by f. Replace f with c=character, i=signed, u=unsigned, x=hex

- If your mouse disappears when you exit DOSBox type CTRL-F10 or Alt-Tab and it should reappear.

- DOSBox has two window sizes: small and full screen Use Alt-Enter to switch between them.

Below are questions you need to answer and submit.

The questions are in a file, toolssub.txt, on the web.
Copy the file .. update the file with the answers ... submit the updated file

```
Submit answers to the questions below.
- toolssub.txt is the file name to submit
- tools        is the submit locker

Instructions.
- Use your browser to save this file as: toolssub.txt
- Use an editor to fill in the answers
- Submit the updated file, toolssub.txt, to the tools submit locker


1. Which software are you using.

   Are you using DOSBox and MASM ...................... Y/N
   If no, what runtime environment are you using ...... ?
   Which operating system are you using
   (XP, Vista-32, Vista-64, Windows 7,
    Windows 8, Linux, other) ......................... ?

2. Approximately, how long did you spend on the TOOLS assignment.

   Answer = ?? minutes.

3. In TOOLS PART-B question 3 you are asked to fill in the absolute
   addresses of the Stack, Data and Code Segments. What were those values.

   --> Remember that segment addresses are always 5 hex digits <--

   Answer = Stack Segment = ?????
            Data  Segment = ?????
            Code  Segment = ?????

4. Run the "TESTKEYS" program described in Part C of the assignment.
   What hex values are generated when you press these keys.

   One byte hex value generated when you press the Enter key       = ??
   Two byte hex value generated when you press the F1 Function key = ?? ??

5. Optional.  Please provide any comments, suggestions,
   constructive criticisms relating to this TOOLS assignment.
```