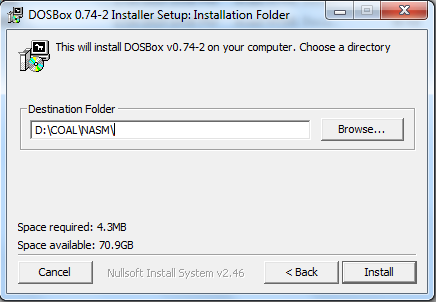
**COAL (E) – Fall 2020**

**Lab Manual 1 - Introduction**

**Activity 1: Setup**

Make a separate folder COAL and NASM in your machine for example “D:\COAL\NASM”. Visit the link given below. Download and install NASM, AFD and DOSBOX, according to the instructions, in your NASM folder.

<http://wetolearn.blogspot.com/2013/09/setting-up-afd-nasm-and-dosbox-for-8086.html>



After installations double click “DOSBox 0.74-2 Options.bat” file and at the end of the file paste following lines:

|  |
| --- |
| MOUNT C D://COAL//NASM  C: |

(We are mounting C drive to our folder where we have saved AFD and we will save our .asm file in this directory)

You may follow links given below for **Video Tutorial of Setup:**

Tutorial part 1: <https://drive.google.com/file/d/1N3lWL8hsN0ZbhF3tlNwCWWwjJ_eHQqk6/view?usp=sharing>

Tutorial part 2: <https://drive.google.com/file/d/10p8qyaOVOwF5lDighrMKE-uNYQX-c3bL/view?usp=sharing>

AFD: <https://drive.google.com/file/d/1eXnD3JEwBelFiJT6iIk7gluudV2Fu_iX/view?usp=sharing>

NASM: <https://drive.google.com/file/d/1ZoeE2MxjNaK7DdJKCacYfAJyn006MI_F/view?usp=sharing>

Dosbox: <https://drive.google.com/file/d/1DnaDIk4RoGBFDP1y4Dr3q89xwM3gx1d1/view?usp=sharing>

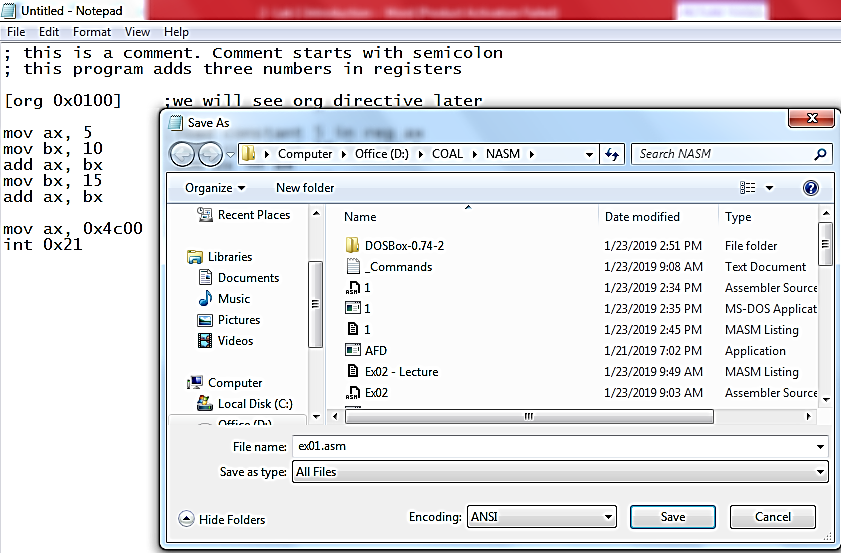
**Activity 2: Running your First Program**

Follow these step in order to run your first program:

1. Copy/paste following code in notepad

|  |
| --- |
| ; this is a comment. Comment starts with semicolon  ; this program adds three numbers in registers  [org 0x0100] ;we will see org directive later  mov ax, 5 ; AX = 5  mov bx, 10 ; BX = 10  add ax, bx ; AX = AX + BX  mov bx, 15 ; BX = 15  add ax, bx ; AX = AX + BX  mov ax, 0x4c00 ;terminate the program  int 0x21 |

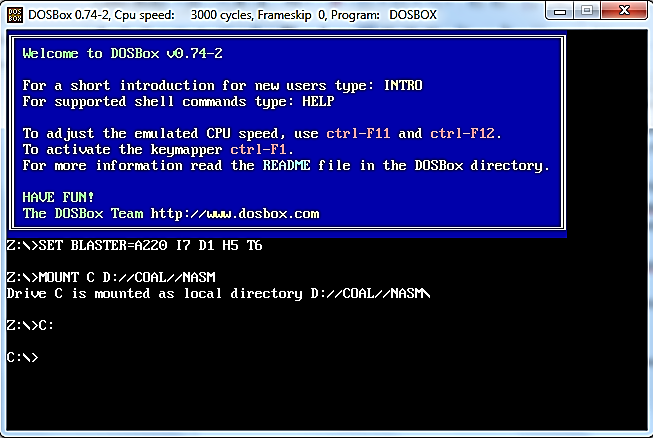
1. Save this file as “ex01.asm” in your NASM folder e.g. “D:\COAL\NASM”:



1. Go to NASM installation directory ( e.g. “D:\COAL\NASM”). Double click **nasmpath.bat** (batch file) and type following command there. (Your .asm file and nasm should be in one folder)

**nasm ex01.asm -o ex01.com -l ex01.lst**

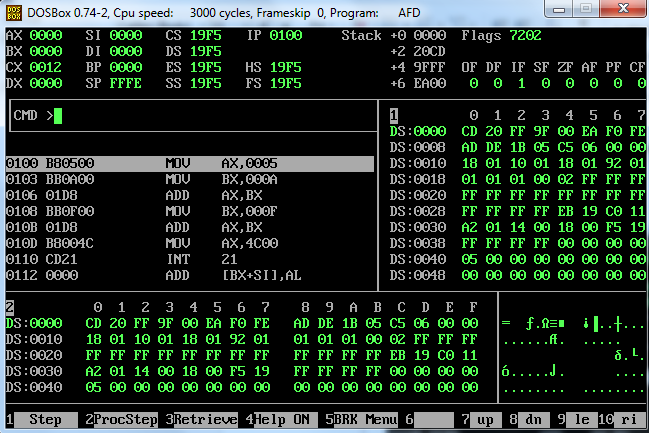
1. Above command will assemble your code and create ex01.com and ex01.lst files. Open ex01.lst file in notepad.
   1. What is opcode of instruction “mov ax, someConstant”
   2. Verify the above opcode everywhere the instruction has been used.
   3. What does “B80500” mean?
   4. Verify the opcode of instruction “mov bx, someConstant” throughout the machine code.
   5. What is the offset of first instruction?
   6. Why are offsets of second and third instructions 3 and 6?
   7. What should be the size of ex01.com file?
   8. Right click ex01.com and verify its size.
2. Open DOSBox (by double clicking dosbox.exe), following window will appear:



1. Write following command and press enter.

**Afd ex01.com**

(Your AFD.exe should be in same directory where we have installed everything)



1. Above command will open the debugger and load your ex01.com file in it.
   1. What is the value of IP register? And what will be its effect?
   2. Note the initial values of data registers
   3. Press F1 and watch the values of data registers

**Activity 3: Explore different functions available in debugger** (after completing activity 4 and 5).

**Activity 4:** Modify this program to generate the sum of first five entries of table of 3, using registers, and watch its execution in the debugger.

**Help:** [Approach 1] Can you do this using two registers only? [Approach 2] Can you do this using one register only if we have **add ax, 3** available in our instruction set? Try both of these approaches and watch the first five entries of table of 3 in AX.

**Activity 5:** Write a program that rotates the value of three registers clockwise twice i.e. given these initial values: ax=10, bx=20, cx=30.

**Help:** First draw data registers on paper. Solve this problem on paper then write your code accordingly. After writing your code verify the execution on paper first then check the execution on AFD. You may use DX.

**Activity 6:** Write a program in assembly language that calculates the square of six by adding six to the accumulator six times.

**Practice Problems**

1. Listing files of two programs are given below. What will be the size of their com files? (\_\_\_\_\_\_\_\_\_\_\_\_)16 and (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)10

Listing File 1:

|  |
| --- |
| 1 ; a program to add ten numbers  2  3 [org 0x0100]  4  5 00000000 BB[1D00] mov bx, num1  6 00000003 B90A00 mov cx, 10  7 00000006 B80000 mov ax, 0  8  9 00000009 0307 l1: add ax, [bx]  10 0000000B 81C30200 add bx, 2  11 0000000F 81E90100 sub cx, 1  12 00000013 75F4 jnz l1  13  14  15  16 00000015 A3[3100] mov [num1], ax  17  18 00000018 B8004C mov ax, 0x4c00 ;terminate the program  19 0000001B CD21 int 0x21  20  21 0000001D 010002000300040005- num1: dw 1, 2, 3, 4, 5, 1, 2, 3, 4, 5  22 00000026 000100020003000400-  23 0000002F 0500 |

Listing File 2:

|  |
| --- |
| 1 ; Multiplication  2  3 [org 0x0100]  4  5 00000000 E90600 jmp start  6  7 00000003 4E9D a: DW 0x9D4E  8 00000005 C3A54E9D b: DD 0x9D4EA5C3  9  10 00000009 B8A9FC start: mov ax, 0xFcA9  11 0000000C C1E004 shl ax,4  12 0000000F C1C004 rol ax,4  13  14 ;mov [a],0x9D4E ; Error op size not specified  15 00000012 C706[0300]4E9D mov word[a],0x9D4E  16  17 00000018 C026[0300]04 shl [a],4 ; by default this will operate on 1 byte  18 0000001D C006[0300]04 rol [a],4  19  20 00000022 C706[0300]4E9D mov word[a],0x9D4E  21 00000028 C126[0300]04 shl word[a],4  22 0000002D C106[0300]04 rol word[a],4  23  24 00000032 B8004C mov ax, 0x4c00 ;terminate the program  25 00000035 CD21 int 0x21 |