Computer Organization and Assembly Language Programming Lab Manual 1
Setup and Running your first program

Activity 1: Setup DOSBOX, NASM and AFD

There are three methods of DOSBOX Setups. You are advised to learn first two.

First method of DOSBOX Setup:

There is a GitHub package that automatically associates Notepad++ with x86 Assembler. Follow these Steps:

1. Follow this link to get started: https://github.com/ASD0x41/Assembly-Programming-Package
2. After clicking on 

button, Download the ZIP file.
3. Extract the folder in an empty directory for ease of use.
4. Open the folder and extract any version of Assembly programmer (dark or Light) in same folder.
5. Click on the extracted folder, double-click on Code file.

There are some shortcuts and instructions given in the code that needs to be followed to run AFD.

Second method of DOSBOX Setup:

A resource zip folder containing DOSBOX , Nasm and AFD has been given along with manual.

1. Follow these Steps:
2. Download the rar file attached and extract the content of the file.
3. Create a new folder in D: directory with the name Assembly (or anyother).
4. Copy all the contents of the folder COALE to Assembly folder.
5. Open dosbox exe file from dosboxportable folder in Assembly folder.
6. Type these command to redirect the current directory to the directory where nasm and Afd exe file are stored.
7. Mount C D:/Assembly (in this case)(use the path if stores somewhere else).
8. C: and press enter

After completing Activity 1, move on to Activity 2 for running your first program.

Activity 2: Running your First Program

Follow these step in order to run your first program:

1- Copy/paste following code in notepad.

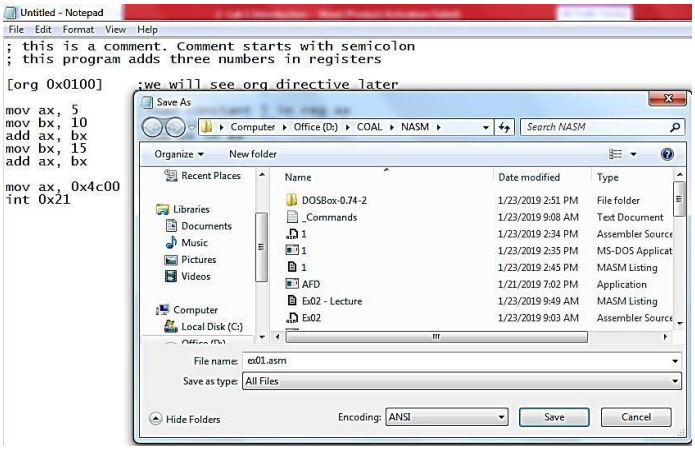
```
; this is a comment. Comment starts with semicolon
; this program adds three numbers in registers

[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
```

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

3- Go to NASM installation directory (e.g. “D:\COAL\NASM”). Double click **nasmpath.bat** (batch file) and type following command there. If there is no .bat file type these commands after opening DOSBOX and mounting it to your NASM directory as told ealier in setup.

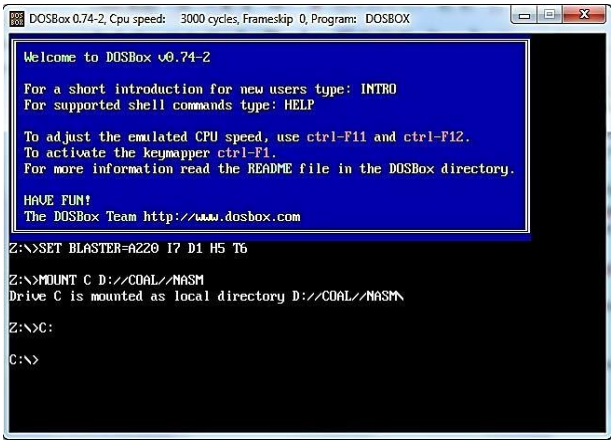
(Your .asm file and nasm should be in one folder)

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



4- Above command will assemble your code and create ex01.com and ex01.lst files. Open ex01.lst file in notepad.

5- Open DOSBox (by double clicking dosbox.exe), following window will appear.

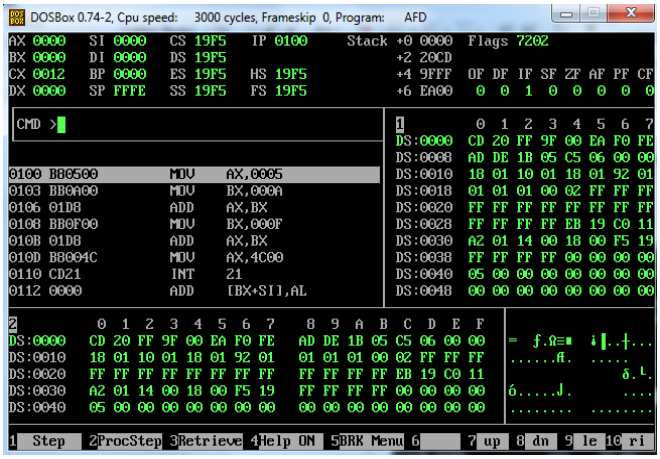


6- Write following command and press enter.

Afd ex01.com

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

7- Above command will open the debugger and load your ex01.com file in it.



LAB TASKS

Attempt following Lab Tasks.

TASK 1:

Answer the followings in a notepad file.

a) Note the initial values of data registers
b) Press F1 and watch the values of data registers
c) What is opcode of instruction “mov ax, someConstant”
d) Verify the above opcode everywhere the instruction has been used.
e) What does “B80500” mean?
f) Verify the opcode of instruction “mov bx, someConstant” throughout the machine code.
g) What is the offset of first instruction?
h) What should be the size of ex01.com file?
i) What is the value of IP register? And what will be its effect?
j) Why are offsets of second and third instructions 3 and 6?
k) Right click ex01.com and verify its size.

TASK 2: Write a program in assembly language that calculates the square of six by adding six to the accumulator six times.

TASK 3:: Do exercise 2 with Byte size operations (for example, AL, AH etc.)

TASK-4: Listing files of a program are given below. What will be the size of its com files? ( )$_{16}$ and ( )$_{10}$

Listing File

```
1                       org 100h
2
3                       start:
4 00000000 BB[1B00]             mov bx, num1
5 00000003 B90A00               mov cx, 10
6 00000006 B80000               mov ax, 0
7
8                       next_add:
9 00000009 0307                 add ax, [bx]
10 0000000B 83C302              add bx, 2
11 0000000E 83E901              sub cx, 1
12 00000011 75F6                jnz next_add
13
14 00000013 A3[1B00]            mov [num1], ax
15
16 00000016 B8004C              mov ax, 4C00h
17 00000019 CD21                int 21h
18
19 0000001B 010002000300040005-     num1 dw 1,2,3,4,5,1,2,3,4,5
19 00000024 000100020003000400-
19 0000002D 0500
```