

## COAL\_A\_p200165\_R2

### Installation of Nasm and DosBox In Ubuntu:

**Assembly Language** is a Low level Language. It is human understandable as compared to machine language. Any Language like **Python, Cpp or Java** ... Or any other language we use all code are converted into machine code because machine can only understand binary even **Assembly code** also converted In to Machine Code.

Assembly Language is a very powerful language. Like You have written some kind of experiment in any high level language that takes week It can happen **Assembly Language** take hours to do that experiment, that's the power of **Assembly Language**.

Another thing if we want to understand the brain of computer then again we have to learn **Assembly Language** because in **Assembly Language** we go Low level and see how the operations are done in machine. In High Level Languages there are built In function and there is a lot of abstraction in them and we unable to see how machine is doing the operations.

In this Lab we have installed the **DOSBOX** and going to run **Hello World** program In machine language. It is not like any other **High level Language Hello World**. We are going to move some data into registers.

First we have installed dosbox and then we have runned our code in dosbox using nasm compiler. First Thing You have to update the repository and then You have to install the **DOSBOX**. To Install **DOSBOX** this is quite simple Just run this command in **Terminal**.

- `sudo apt update`
- `sudo apt install dosbox`

### Nasm and AFD Debugger :

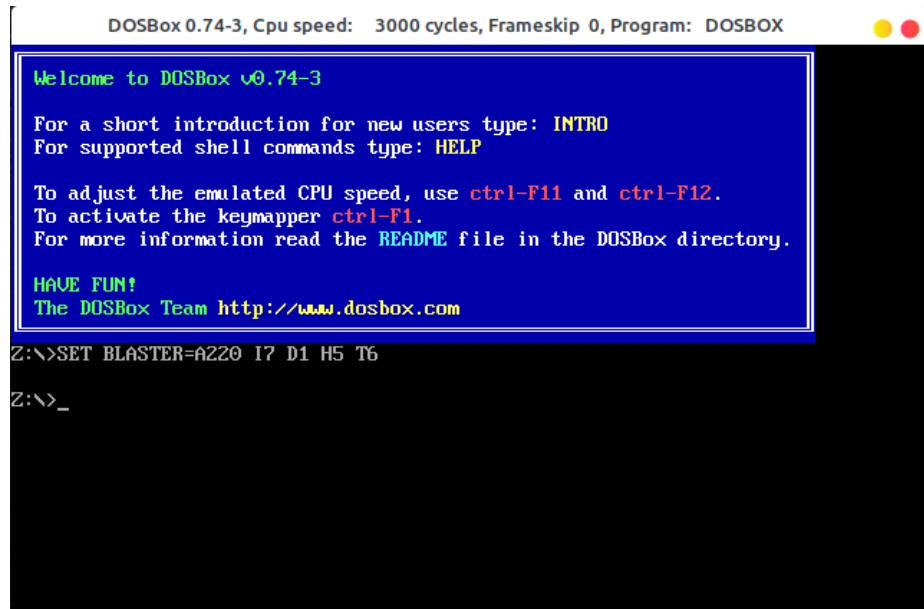
Once that done You can run dosbox using terminal by just typing dosbox in your terminal. but the next thing we need is the nasm compiler and Advance free debugger that will be used to debug your code.

**AssmSoft.zip Folder** is provides us in the lab and extract that folder. Once that you extract it place that files with your assembly code files. One thing to remember is that your code files and **AssmSoft.zip Folder** extracted files must be placed together either you will get error when you are executing your **Assembly Code files**.

# Program that we Runned In Lab:

## Step 1:

Open Up the Dosbox by just typing **dosbox** in your terminal.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

Welcome to DOSBox v0.74-3

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

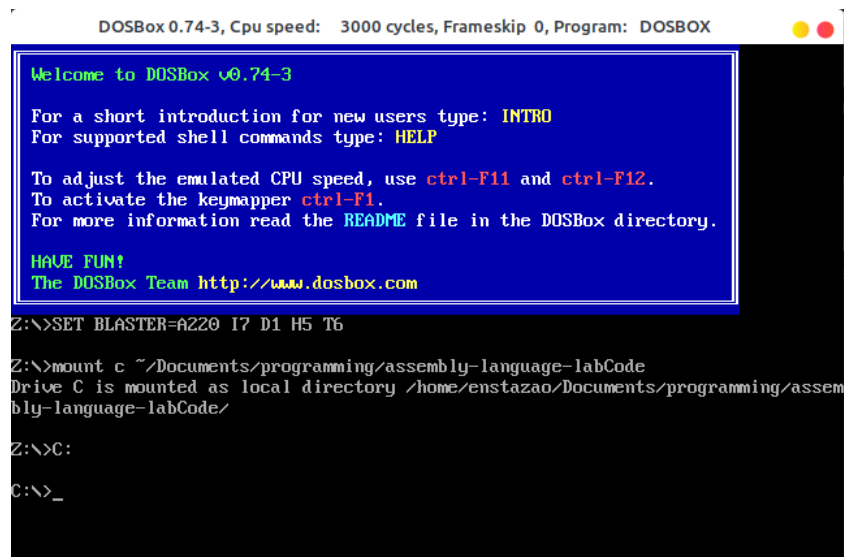
HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>_
```

## Step 2:

Now you have to mount the drive where you have codes and also the extracted files. In my case the path is in **Documents** in your case it may be different just place the correct path while mounting. I am going to mount In **C** drive.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

Welcome to DOSBox v0.74-3

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c ~/Documents/programming/assembly-language-labCode
Drive C is mounted as local directory /home/enstazao/Documents/programming/assembly-language-labCode/

Z:\>C:

C:\>_
```

**Step 3:** Once that step is done Now it's Time to Assemble your code. Just paste this code and name the file code01.asm or you can use any name you want. But when assembling use same name.

### CODE:

```
[org 0x0100]
```

```
; start of code
```

```
mov ax, 5           ; move the constant 5 into register ax
mov bx, 10
```

```
add ax, bx          ; add value of bx into the value of ax
```

```
mov bx, 15           ; add constant 15 into the value of bx
add ax, bx
```

```
mov ax, 0x4c00       ; exit ..
int 0x21              ; .. is what the OS should do for me
```

Once that done Now run this command to assemble your code. If your code have no error then your code will be compiled successfully.

### Command:

**nasm code01.asm -o code01.com**

This command mean that put the output of my code in code01.com file. You will not see any output on screen because we have not printed anything on screen. This output will be seen by you.

You can use **Advance free debugger** to see what the code does. To open the **AFD** you can use this command and you to give .com file as a input not the .asm file.

### Command:

**afd code01.com**

This screen will be seen In your computer. That's All we have done in our second lab.

