**PRACTICAL # 01**

**OBJECT:**

Tools and environment setup for assembly language programming.

**THEORY:**

This course is aimed to cover the basics of Intel 80x86 assembly language programming. The language is important because it is the interface between the software world of high-level languages like C and Java and the hardware world of the CPU. Assembly language is the lowest-level, human-readable programming medium we can use to express complete application programs. Assembly language gives full access to the programmable features of the hardware, so a good understanding of it will provide valuable insight into the fundamentals of CPU design, its operation of and the way of program execution.

The goal of Lab 1 is simply to introduce the basic tools and procedures you will use to write, assemble, link, execute, and debug your programs.

An **assembler** is a program that translates assembly language code into machine language. The **linker** program links the required libraries to the assembled object code (assembler output) to make the code executable.

There are many assemblers available. We will use the MASM assembler. In addition, we will also use an emulator program that has built in assembler. The emulator we'll use is EMU8086.

**Setting up the assemblers**

**MASM Setup**

MASM is the set of tools that contains assembler and linker programs.

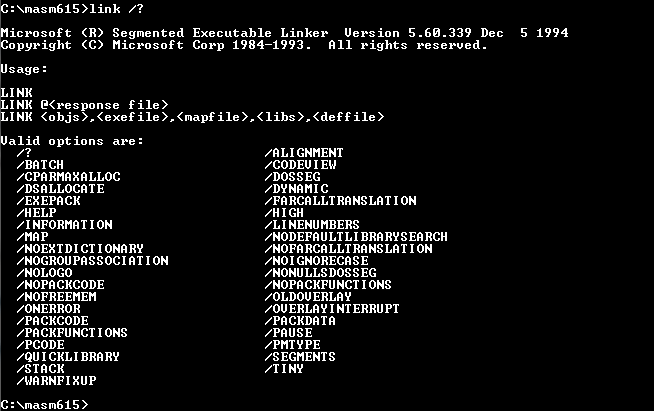
Download the 16 bit MASM as ZIP.

Extract the ZIP file to some suitable location like "c:\masm5", or whatever version you have. This folder can be something like "c:\\masm615" should have subfolders including "bin" and "programs".

**Setting up the environment variable**

Environment variable offers the ease to call some program from any location on the file system. To setup the environment variable for MASM615 click on Start/settings/control panel. Open "System", select the "Advanced" tab, and click on "Environment Variables". In the lower text box, select "path", and click on the Edit button. Add to the righthand end of the text which is then displayed above in "variable value", a semicolon and then: c:\masm615\bin. In case of MASM5 set path to “c:\masm5”.

Now open the command prompt. In DOS window, type cd c:\\masm615\programs. This will make "programs" the active directory. You should place all the Assembler programs that you create in this folder.

The DOS interface for MASM looks something like given in fig 1.

**Fig 1: Linker of the MASM**

To build a program type: “ml myprogram.asm”  
This should produce an executable called "myprogram.exe".  
At the command prompt, type: myprogram.exe  
and, if everything up to this stage is fine, the program should show the output of the program.

To build programs in MASM5 open CMD. Change the current directory to MASM 5 by typing **CD C:\ masm5**. Type **MASM.EXE program.asm** and press enter. It will ask for output file names, just keep pressing Enter to generate default file names. Program.asm is your assembly program source code.

Next type **LINK program.obj**. Program.obj is binary file generated by the assembler MASM.exe.

Keep pressing Enter to generate default file output file names. This generates an executable file called program.exe. Execute this is program in CMD by just typing its name.

**EMU8086 Setup**

Emu8086 combines an advanced source editor, assembler, disassembler, emulator (of 8086 microprocessor) with debugger, and many other features. It includes a pack of sample programs. This program is very helpful for those who just begin to study assembly language. It compiles the source code and executes it on emulator step by step.

Visual interface provides a way to look at the microprocessor while it is executing instructions. It shows registers, flags and memory while your program executes.

To install the emulator program,

Download the setup and execute the setup file to install it. The setup will ask for installation path as shown in fig 2. Provide some suitable path for installation.

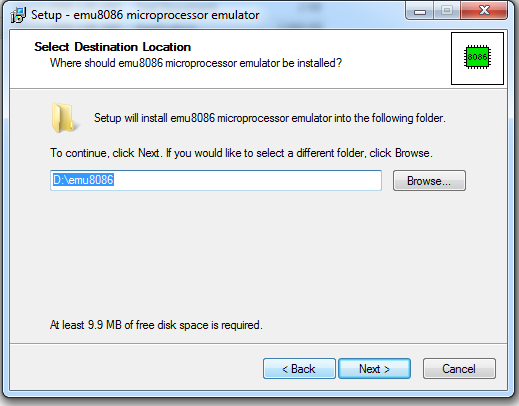
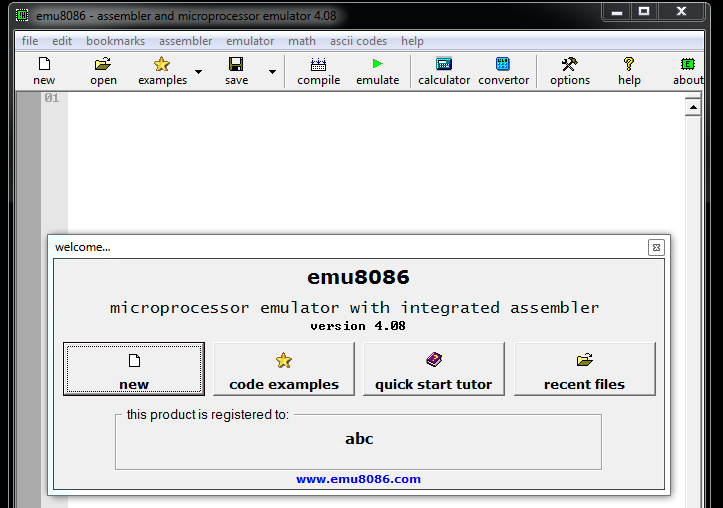


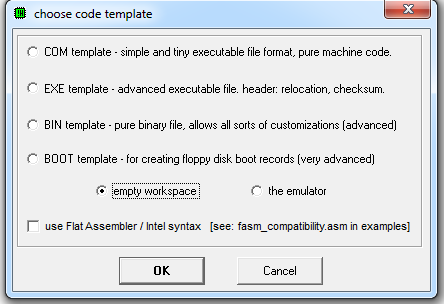
Fig 2: Emu8086 setup installation path

After installation is complete execute the program to get started with assembly language. When the program is started it will show the following screen.



**Fig 3: Emu8086 first execution**

Here you can study the per-packaged code examples. To build your program choose “**new**” button. Next you will be asked to choose a program template. Choose the “**Empty Workspace**” option and press “**OK**”.



**Fig 4: Emu 8086 choosing new empty workspace for assembly programming**

This will open the text editor where you can write your first assembly program, assemble build and run the program.

**Program structure:**

8086 based assembly program has a well-defined program structure. A trivial assembly program consists of three sections; **the stack segment, data segment** andthe **code segment**.

**Stack segment** is the memory area that holds local variables that are created in for different procedures in the program. Local variables are lost when a function/procedure ends. The size of stack is mentioned at the start of the program. For our programs, which will not take much stack space, we will use default stack size of 100 Hex bytes. To do this we write .STACK 100H.

**Data segment** offers memory for the global program data, the constants and uninitialized data variables. The following is a very simple assembly program that defines the basic skeleton of an assembly program. The size of the data segment depends on .MODEL declaration.

**Code segment** contains the executable code of the program. Its size also depends on .MODEL declaration. For SMALL model its typically 64KB.

.MODEL SMALL

.STACK 100H

.DATA

.CODE

MAIN PROC

; code goes here

; The following two instructions are required to return the control to the O.S. They must be written just

; before the main proc ends

mov ah, 4ch

int 21h

MAIN ENDP

END MAIN

Note: The assembly program source files have .asm extension. Comments in assembly programs start with **;** and continue till end of the line.

**ACTIVITIES**

**Activity 1**

Write the above program in a text editor of your choice. Save the program with .asm extension in MASM folder. Select **save as type** as “**All files**”. Now in DOS and navigate to the MASM directory. Assemble the program as described above.

**REVIEW QUESTIONS**

1. What are the main benefits of the assembly language?
2. What is the program called that converts assembly code to machine code?
3. What is the purpose of LINK program (Linker)?
4. What features does Emu8086 offers?