Lab2- Introduction to MASM

2.0 Introduction

Assembly language unlocks the secrets of computers hardware and software. An assembler converts source code to machine own language. Microsoft Macro Assembler (MASM), product of Microsoft Corporation includes several features which makes programming efficient and productive. The following chapter will give an overview how to use MASM for assembling the 80x86 program you code.

2.1. Brief introduction to MASM and how to write ALP and execute ALP

2.2. Using Model Tiny(All the data and code fit in one segment Tiny programs when compiled give .COM executable -the program must be originated at location 100H)

```
.Model Tiny
.386                                    ;Assembler by default accepts only
                                        8086/8088 instructions, unless a
                                        program is preceded by .386/.486
                                        directive to select the microprocessor
.data                                   ; Data Segment
COUNT       EQU     32H
VAL1        EQU     0030H
DAT1        DB      45H,67H,100,'A'
WRD         DW      10H, 3500H,0910H
DAT2        DD      0902H
DAT3        DW      2 DUP(0)
DAT4        DB      56H
RES         DB      10 DUP(?)
DWRD        DD      01020304H
.CODE
.STARTUP
            MOV     SI,DAT3
            MOV     AL, DAT1 + 1
            MOV     BX,WORD PTR DAT1+4
            ADD     BX,20H
            MOV     AI,[BX]
            LEA     BX,DAT4
            MOV     AL,[BX]
            MOV     BX,VAL1
            MOV     AL,FS:[BX]
            MOV     EBX, DWRD
.EXIT
END
```

2.3 Assembling (Tell them how to assemble)

There are two methods of assembling

Method 1:

Type MASM filename.asm <enter>
If no error in code there is .OBJ file is generated
Now type
LINK filename.obj <enter>
Check the files created at each step and examine the content of .lst and .map file

Method 2:

Typ'e ML filename.asm <enter>
To create list and map file command format is 'ml /Fl /Fm Filename.asm
Check the files created at each step and examine the content of .lst and .map file
What is the difference between Method 1 and Method 2?
To check the working of the program – execute DEBUGX Filename.com and then trace or go in DEBUGX.

Note:

1. All your files .asm, .com/.exe must be present in MASM611/BIN
2. Make sure that you copy debugx.exe into MASM611/BIN folder

Tasks:

1. Write an ALP to add two 5 byte data stored at location dat1 & dat2. Now store the result of the addition at location dat2 + 20d.

2. Write an ALP that will transfer data from 'ARRAY1' to 'ARRAY2'. The number of elements in the array is 50. The array is a byte array. The starting address of ARRAY2 = starting address of ARRAY1 + 60d.

Lab2 Plan:

1. **Tell them how to use MASM**
2. **Then tell them to insert the given code, what should be the extension(allow them to copy the code....if they want to copy)**
3. **Tell them how to assemble**
4. **Then, how to check the working of ALP**
5. **Finally, assign Task 1 and then Task 2. (Tell them to follow the sequence and show the results)**

Note:Instruct them to use small numbers ...during data declaration...so that there will be no carry.