MASM-Microsoft Macro Assembler

- To download dosbox,use the following link

https://www.dosbox.com/download.php?main=1

- To download 8086 zip file use the following link

http://www.mediafire.com/file/mm7cjztce9efj4w/8086.zip/file

Note: After downloading both, save into c drive

1. Install DosBox directly into your C Drive.
2. Extract files from ZIP folder, it will contains multiple exe files. Place this folder into your C drive and rename it to 8086(if its name is not 8086)
3. Run DosBox and write the following:

mount c c:\8086

4. Create a file with .asm extension e.g. hello.asm using notepad in 8086 folder.

Syntax of MASM programming

```
ASSUME CS : CODE , DS : DATA
DATA SEGMENT
 sum. db 10H
      :
DATA ENDS

CODE SEGMENT
START : MOV AX, 2000H
        MOV DS, AX
          :
          :
CODE ENDS
END START.
```

Write a MASM program to display “Hello World”

```
data segment
str db 0ah,0dh,"Hello World$"
data ends

code segment
assume cs:code,ds:data
start:
mov ax,data
mov ds,ax
mov dx,offset str
mov ah,09h
int 21h
mov ah,4ch
int 21h
code ends
end start
```

To make your directories available as drives in DOSBox by using the "mount" command

Command for assembling

```
C:\>masm hello1.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [hello1.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51708 + 464836 Bytes symbol space free

       0 Warning Errors
       0 Severe  Errors
```

Command for link

Command for execution

Write a MASM program to add two numbers(nos initialized in the program itself)

```
data segment
  opr1 dw  1234h
 opr2 dw  0023h
 result dw 01 dup(?)
data ends
code segment
start: mov ax, data
       mov ds ,ax
       mov ax, opr1
       mov bx, opr2
       clc
       add ax,bx
```

- mov di, offset result
- mov [di],ax
- mov ah,4ch
- int 21h

```
code ends
end start
```

DOS function calls

- By **calling** INT 21h with a sub function number in the AH processor register and other parameters in other registers, various **DOS** services can be invoked.
- These include handling keyboard input, video output, disk file access, program execution, memory allocation, and various other activities.

```
Mov ah, 01h
int 21h      (read char from std input device)

Mov ah, 02h
int 21h  (Write char to std device)

LEA dx,str                  // ( str db oah,odh, “Hello$”)//
Mov ah, 09h
int 21h                     //(display string)//
```

```
LEA dx,str                  // ( str db 24h dup(?) )//
Mov ah, 0ah
int 21h                     // (Buffered str input)//

Mov ah, 4Ch
int 21h (Program to exit)
```

Syntax of MACRO

i) Defining a Macro

MACRO can be defined anywhere in a program using derivatives MACRO and ENDM

Eg: display macro

```
        mov dx,offset str
    mov ah,09h
    int 21h
endm
```

(for invoking this in program just write display)

- ii) Passing parameter to macro

```
display macro msg
            mov dx,offset msg
          mov ah,09h
         int 21h
Endm
```

(for invoking it
Display opr1
Display opr2
Display result)