Lecture Title

Course Code: 0052

Course Title: Computer Organization and Architecture

Dept. of Computer Science
Faculty of Science and Technology

Lecturer No:	**4(b)**	**Week No:**	**5**	**Semester:**	
Lecturer:					

Overview

1. Creating, Assembling and executing assembly language program.

2. By the end of this lesson we will be able to write simple but interesting assembly program.

Program Structure

- A program Consist of
  - **Stack**
  - **Data**
  - **Code**
- Each part occupies memory segments
- Program segment is **translated** into memory segment by assembler.
- The size of code and data of a program can be specified by **memory model** using **.MODEL** directive

.MODEL Memory_model

.MODEL SMALL [Code in ONE segment and Data in one segment]

Stack Segment

- Allocate a block of memory (stack area) to store the stack.
- The stack area should be big enough to contain the stack at its maximum size.
- **Declaration:**

```
.STACK     size

.STACK     100H
```

** Allocates 100 bytes for stack area reasonable size for most applications

** If size is omitted 1KB is allocated for stack area.

Data Segment

- Contains all the **variable** definitions and sometimes Constant definitions (constant does not take any memory).
- To declare data segment **.DATA** directive is used followed by variable and constant declaration.

```
.DATA

WORD1      DW   2

BYTE1      DB   1

MSG        DB   'THIS IS A MESSAGE'

MASK       EQU  10010001B
```

Code Segment

- Contains the program’s instructions
- **Declaration:**
- **.CODE** name [name is optional]

  There is no need of **name** in SMALL program
- Inside a code segment, instructions are organized as procedures.

  name PROC

  ; body of the procedure

  name ENDP
- Here name = name of the procedure. PROC and ENDP are pseudo-ops

Program Structure

```
.MODEL      SMALL

.STACK      100H

.DATA

; data definitions here

.    CODE

     MAIN PROC

          ;instructions go here

     MAIN ENDP

;other procedures go here

END MAIN
```

*** The last line of the program should be the END directive, followed by the name of main procedure

Instruction: INT (Appendix C)

- **INT**: Interrupt option stops the continuous progress of an activity or process.
- **Syntax:**

```
INT  interrupt_number
```

***A particular function is requested by placing a function number in the **AH** register and **invoking INT 21h** .

*** **INT 21h** functions expect input values to be in certain registers and return output values to other registers

Function Number	Routine	Input	Output
1	single-key input	AH=1	AL = 0 if no input or ASCII of character
2	single-character output	AH=2	DL=ASCII of display char AL= ASCII of display char
9	character-string output	AH=9	

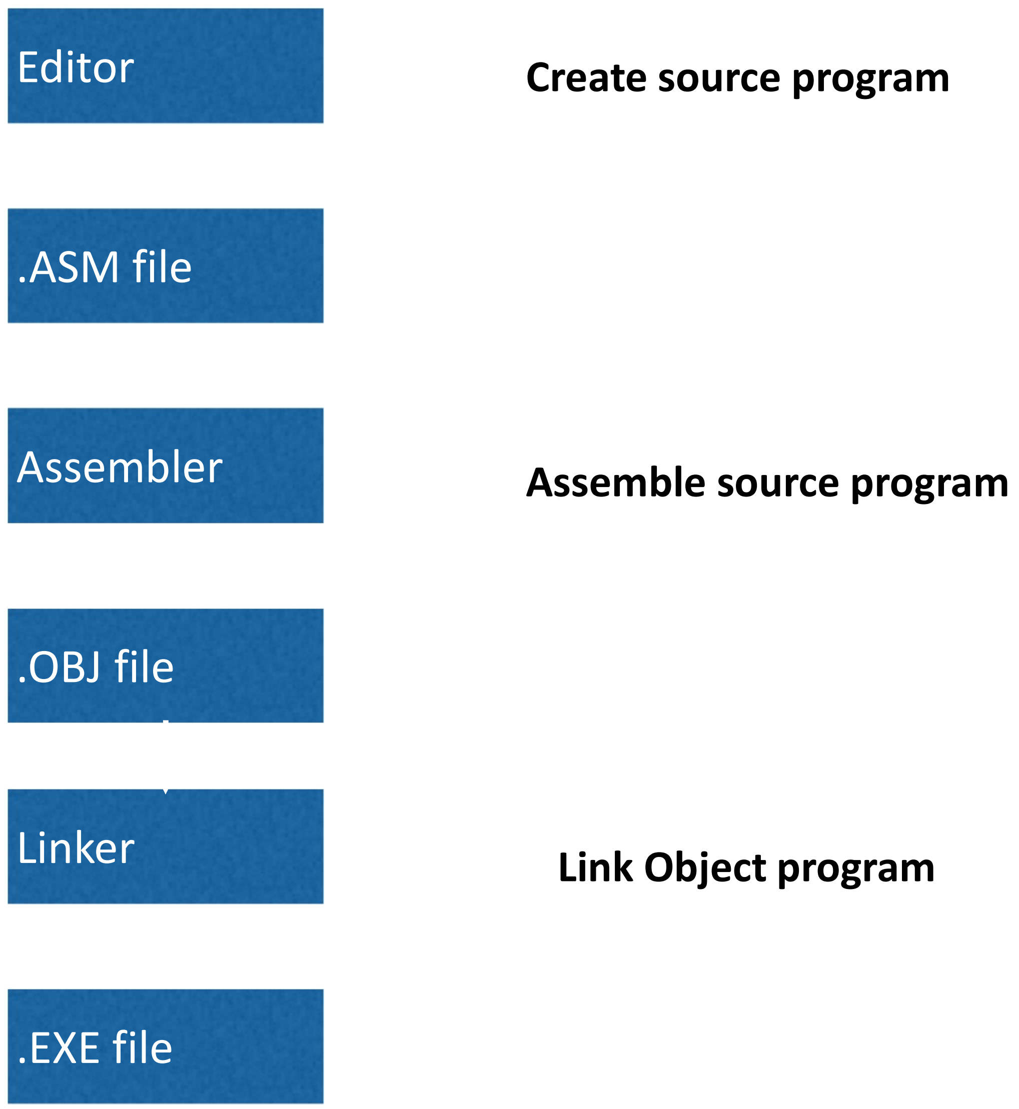
The First Program

- Task: The program will read a character from the keyboard and display the same at the beginning of next line.
- Lets start by displaying a question (“?”) mark for the user input

The Solution

```
.MODEL      SMALL
.STACK      100H
.    CODE
 MAIN PROC
; display prompt to the user
 MOV AH,2 ; display character function
 MOV DL,'?' ; character is '?'
 INT 21H     ; display the DL char (?)
;input a character
 MOV AH,1   ; read character function
 INT 21H     ; character is in AL
 MOV BL,AL ; save input to BL reg
;go to new line
   MOV AH,2        ; display character function
   MOV DL,0Dh     ; carriage return
   INT 21H       ; execute carriage return
   MOV DL,0AH    ; line feed to display
   INT 21H       ; execute Line feed
   ; display character
   MOV DL, BL      ; retrieve character
   INT 21H
   ;return to DOS
   MOV AH,4CH  ; terminate the currant process and
transfer
                         control to invoking process
   INT 21H         ; termination the execution of
program
                   return control to DOS
   MAIN ENDP
   END MAIN
```

Programming Steps



Instruction: LEA

- LEA: Load Effective address

  LEA destination, source
- LEA puts copy of the source offset address into the destination.

  i.e. LEA DX, MSG   ; will load address of MSG to DX

Program Segment Prefix (PSP)

- PSP contains information about the program to facilitate the **program access** in this area
- DOS places its segment number in both DS and ES before program execution
- Usually, DS does not contain the segment number of the data segment.
- Thus, a program with data segment will start with these two instruction

MOV AX,@DATA [name of data segment define in .DATA]

MOV DS,AX

Solve the Following

1. **Write a program to print HELLO! on the screen**
2. **Write a program that can convert the user input character in UPPERCASE like below**

```
ENTER A LOWER CASE LETTER:  a
IN UPPERCASE IT IS:  A
```

References

- Assembly Language Programming and Organization of the IBM PC, Ytha Yu and Charles Marut, McGraw Hill, 1992. (ISBN: 0-07-072692-2).
- https://www.tutorialspoint.com/assembly_programming/index.htm

Books

- Assembly Language Programming and Organization of the IBM PC, Ytha Yu and Charles Marut, McGraw Hill, 1992. (ISBN: 0-07-072692-2).

- Essentials of Computer Organization and Architecture, (Third Edition), Linda Null and Julia Lobur

- W. Stallings, “Computer Organization and Architecture: Designing for performance”, 67h Edition, Prentice Hall of India, 2003, ISBN 81 – 203 – 2962 – 7

- Computer Organization and Architecture by John P. Haynes.