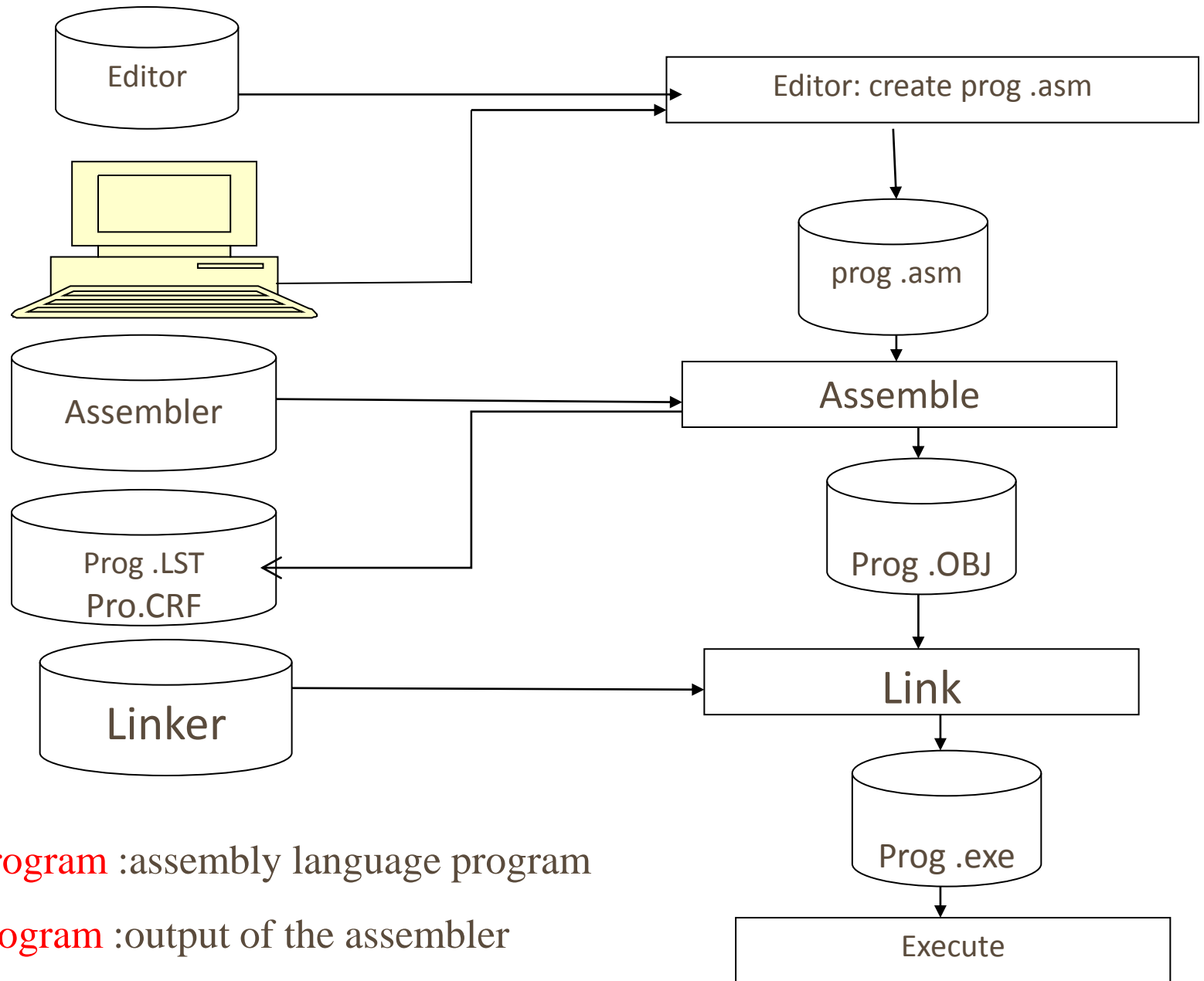


Executing and Linking an assembly program

Dr . Shahenda sarhan

Lecturer @ CS Department

Steps in Assembly , Link and execute



Source Program :assembly language program

Object Program :output of the assembler

Executable Program : output of the linker

The steps of preparing a program for execution:

- **Assembling**

- The assembly steps involves translating the source code into object code and generating an intermediate .OBJ file or module.
- The assembler calculates the offset for every data item in the data segment and every instruction in the code segment.
- It also creates a header immediately ahead of the generated .OBJ module. Part of the header contains information about incomplete addresses.

- **Linking**

- Convert the .OBJ module to an .EXE machine code module.
- Combine separately assembled programs into one executable module.

- **Loading**

- load the program for execution.
- complete any addresses indicated in the header that were left incomplete.
- drops the header and creates a PSP immediately before the program loaded in memory.

Quiz

- ADD AX,BX
- SUB AX,BX
- MUL BX
- DIV BX

Using the previous instructions calculate

1. $X^2 + Y + XY$

2. $X+Y$

3. XY/ Y

4. $X+Y+Z$

Example 1

; Add two numbers and store the results into the third variable

page 60,132 page [length(10-255)],[width(60-132)]

TITLE A04ASM1 (EXE) Move and add operations

; -----

STACK SEGMENT PARA STACK 'Stack'

DW 32 DUP(0)

STACK ENDS

; -----

DATASEG SEGMENT PARA 'Data'

FLDD DW 215

FLDE DW 125

FLDF DW ?

DATASEG ENDS

; -----

CODESEG SEGMENT PARA 'Code'

MAIN PROC FAR

ASSUME SS:STACK,DS:DATASEG,CS:CODESEG

MOV AX,DATASEG ;Set address of data

MOV DS,AX ;Segment in DS

MOV AX,FLDD ;Move 0215 to AX

ADD AX,FLDE ;Add 0125 to AX

MOV FLDF,AX ;Store sum in FLDF

MOV AX,4C00H ;End processing

INT 21H

MAIN ENDP ;End of procedure

CODESEG ENDS ;End of segment

END MAIN ;End of program

Example 2 :Printing ASCII code

```
.Model Small
.Stack 100h
.Code
MAIN PROC
    MOV AH , 2
    MOV CX , 256
    MOV DL , 0

Print_Loop:
    INT 21h ;print the character in DL
    INC DL ; next character
    DEC CX ; decrement
    JNZ PRINT_LOOP ; if cx ≠0 jmp to print_loop
    MOV AH , 4Ch
    INT 21h

MAIN ENDP
END MAIN
```

Example 3

CASE AX

< 0 : PUT -1 IN BX

= 0 : PUT 0 IN BX

> 0 : PUT 1 IN BX

END_CASE

CMP AX, 0 ; check AX

JL NEGATIVE ; AX < 0

JE ZERO ; AX = 0

JG POSITIVE ; AX > 0

NEGATIVE :

MOV BX,-1

JMP END_CASE

ZERO :

MOV BX,0

JMP END_CASE

POSITIVE :

MOV BX,1

END_CASE:

Example 4

- Compare two numbers to get the bigger

Below: `CMP AX,BX`

`JG Below`

Example 5

- Divide 160 by 4

`MOV AX, 160`

`MOV CL,2`

`SHR AX, CL`

INT 21

- Is a DOS function used to perform some tasks while the value of the needed service is placed in AH
- 1 : read a value from keyboard
- 2: write a letter on the screen
- 9: write a string on the screen

Example 6 : to read a char from keyboard the print it on the screen in the beginning of the next line

```
.MODEL SMALL
```

```
.STACK 100H
```

```
.CODE
```

```
MAIN PROC
```

```
MOV AH,2 ; طباعة حرف
```

```
MOV DL,'?' ; الحرف المطلوب طباعته
```

```
INT 21H
```

```
; قراءة حرف من لوحة المفاتيح
```

```
MOV AH,01 ; قراءة حرف
```

```
INT 21H
```

```
MOV BL,AL ; تخزين الحرف
```

```
MOV AH,02
    MOV DL,0DH    ; carriage return
    INT 21H
    MOV DL,0AH ; line feed
    INT 21H
    MOV DL,BL    ; ; إحضار الحرف من المسجل
    INT 21H
    MOV AH,4CH
    INT 21H
MAIN ENDP
END MAIN
```

Chapter 7

Differences between .EXE and .COM

1. Program Size

- .COM programs only have 1 segment for both instructions and data
- restricted to a size of 64K or less
- always smaller than the same program assembled as a .EXE file

2. Segments

- .COM combines PSP, stack, data, and code segments into one segment
- Assembler automatically generates a stack for a .COM program

3. Initialization

- The loader will load a .COM program but will initialize all segment registers to the start of the PSP
- Programmer must include a ORG 100H directive to get the registers to point to the right place

TITLE A05COM1 COM program to move/add

CODESEG SEGMENT PARA 'Code'

ASSUME CS:CODESEG, DS:CODESEG, SS:CODESEG,
ES:CODESEG

ORG 100H

BEGIN JMP MAIN

;

FLDD DW 215

FLDE DW 125

FLDF DW ?

;

MAIN PROC NEAR

MOV AX, FLDD

ADD AX, FLDE

MOV FLDF, AX

MOV AX, 4C00H

INT 21H

MAIN ENDP

CODESEG ENDS

END BEGIN

