Mansoura University
Faculty of Computers and Information
Department of Computer Science
First Semester: 2020-2021

[CS214P] Assembly Language: Chapter 5

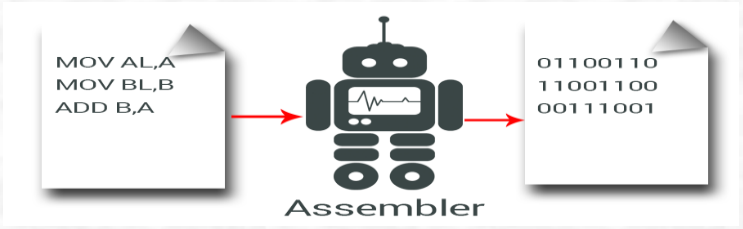
Grade: Third Year (Computer Science)

Sara El-Metwally, Ph.D.

Faculty of Computers and Information,

Mansoura University,

Egypt.



Computer Science Department
Faculty of Computers and Information
Mansoura University

Assembly Language:

Assembling, Linking, and Executing Programs

Sara El-Metwally, Ph.D.
Faculty of Computers and Information,
Mansoura University, Egypt.

Email: sara.elmetwally.2007@gmail.com
Office: Faculty of CIS, third floor

Compiler

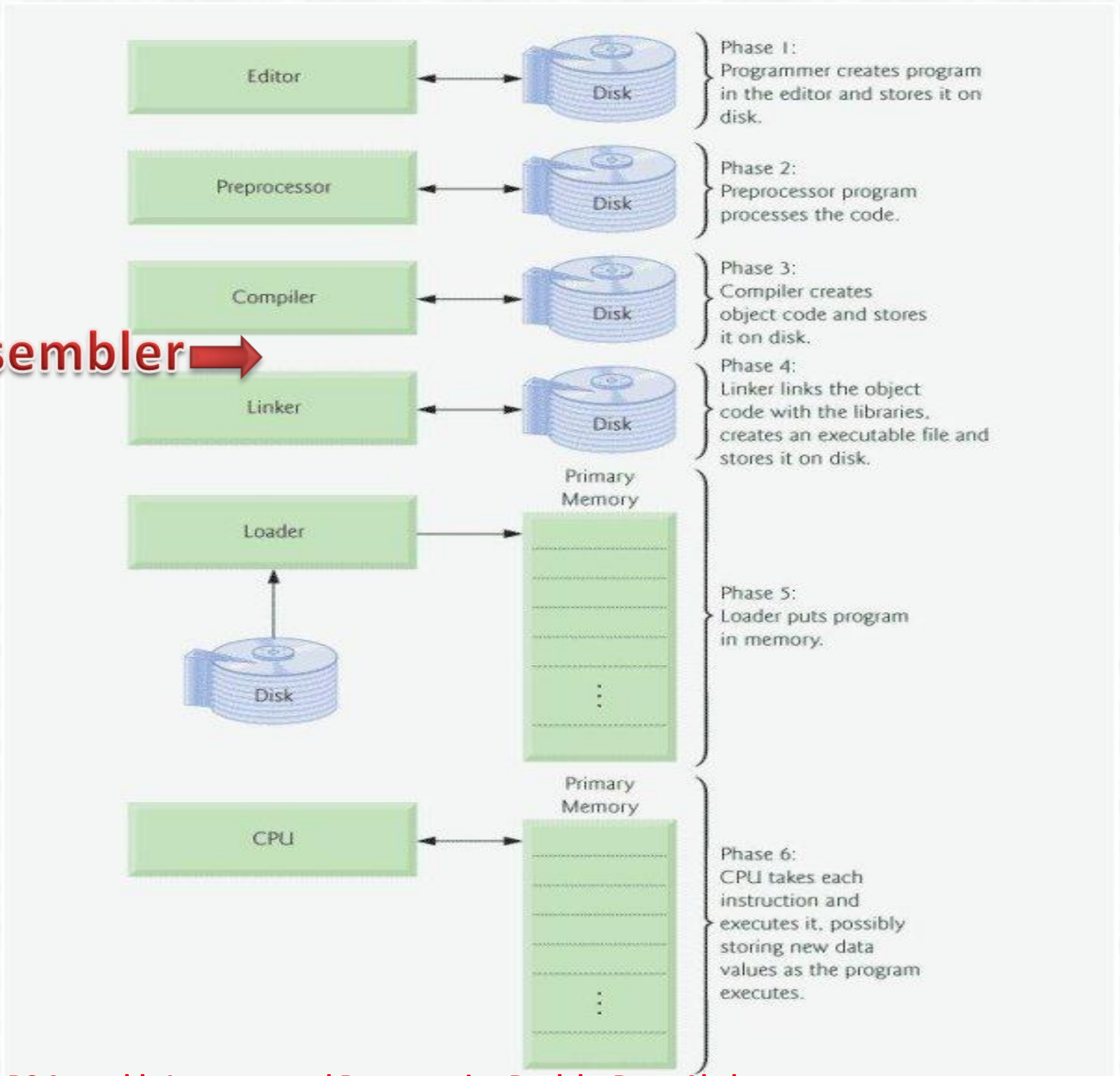
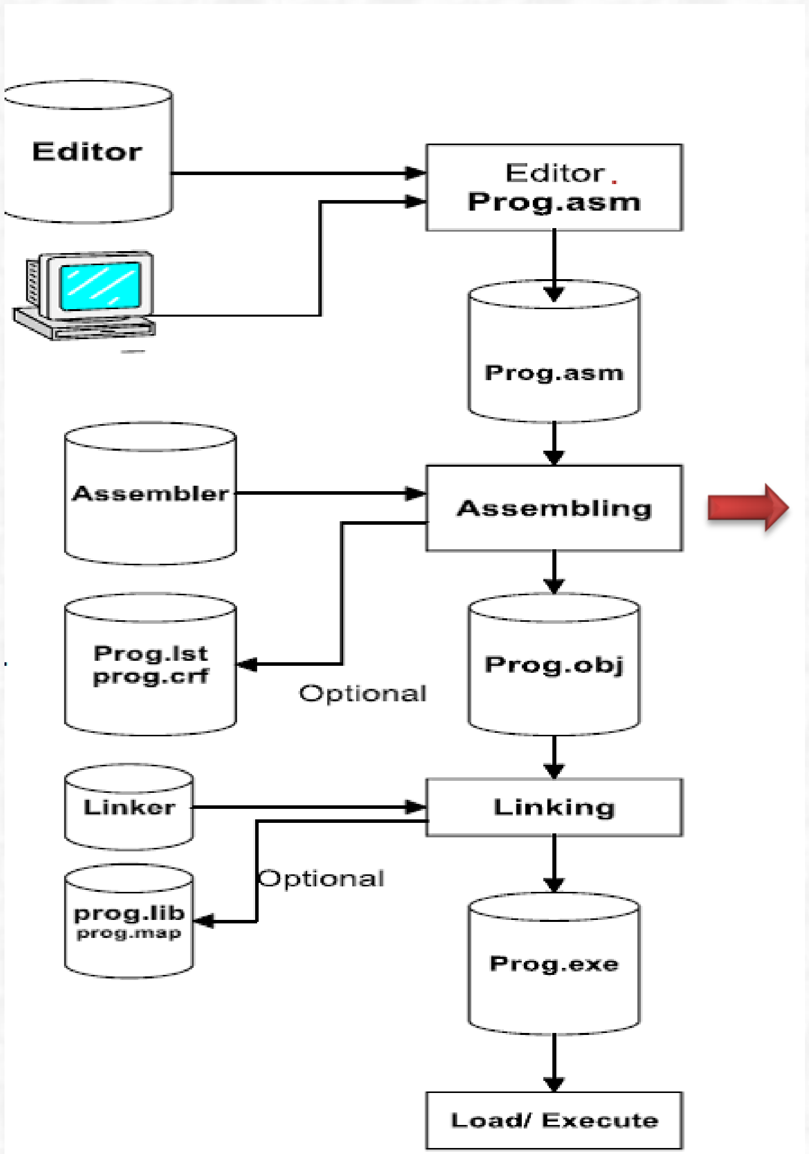

Editor
Disk
Phase 1:
Programmer creates program in the editor and stores it on disk.
Preprocessor
Disk
Phase 2:
Preprocessor program processes the code.
Compiler
Disk
Phase 3:
Compiler creates object code and stores it on disk.
Assembler
Linker
Disk
Phase 4:
Linker links the object code with the libraries, creates an executable file and stores it on disk.
Loader
Disk
Primary Memory
Phase 5:
Loader puts program in memory.
CPU
Primary Memory
Phase 6:
CPU takes each instruction and executes it, possibly storing new data values as the program executes.

Image credit: IBM PC Assembly Language and Programming Book by Peter Abel

Assembling, Linking and Executing Programs

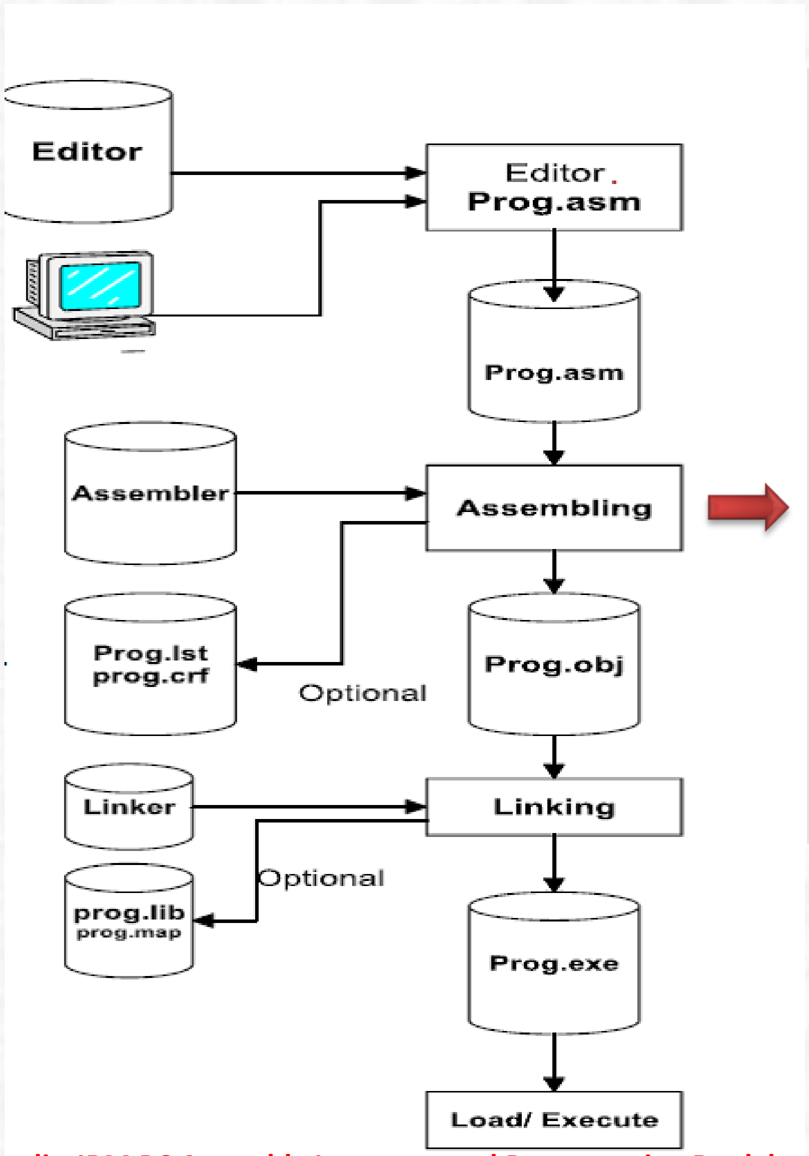


Assembler 1st Pass

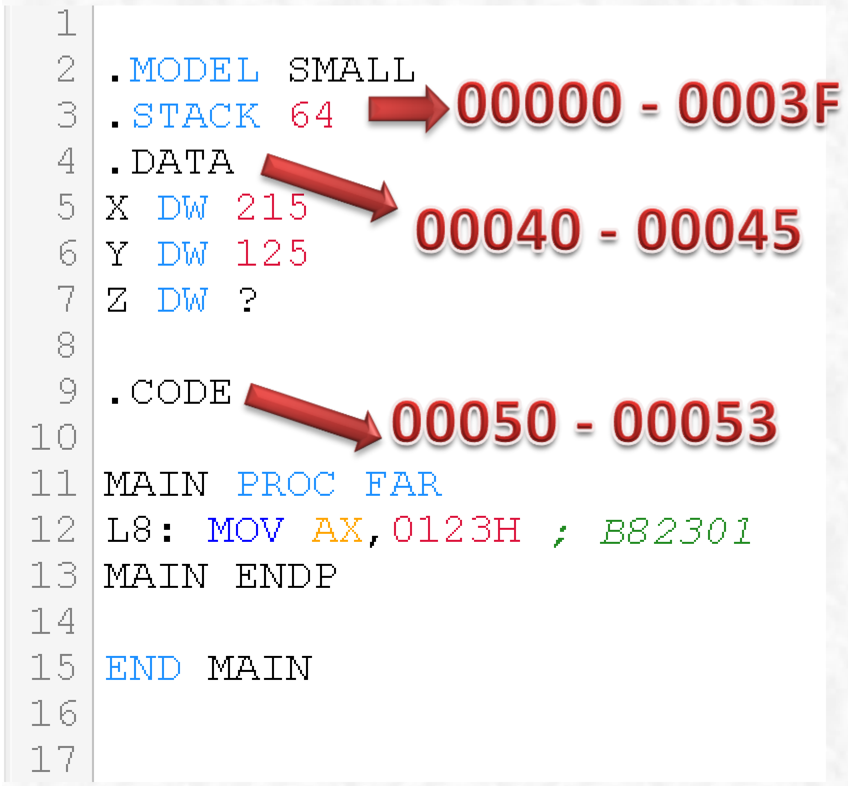
- Assembler reads the source code and construct a symbol table of names, labels used in the program and their relative location within segment.
- Assembler determines the amount of code to be generated for each instruction.

Image credit: IBM PC Assembly Language and Programming Book by Peter Abel

Assembling, Linking and Executing Programs



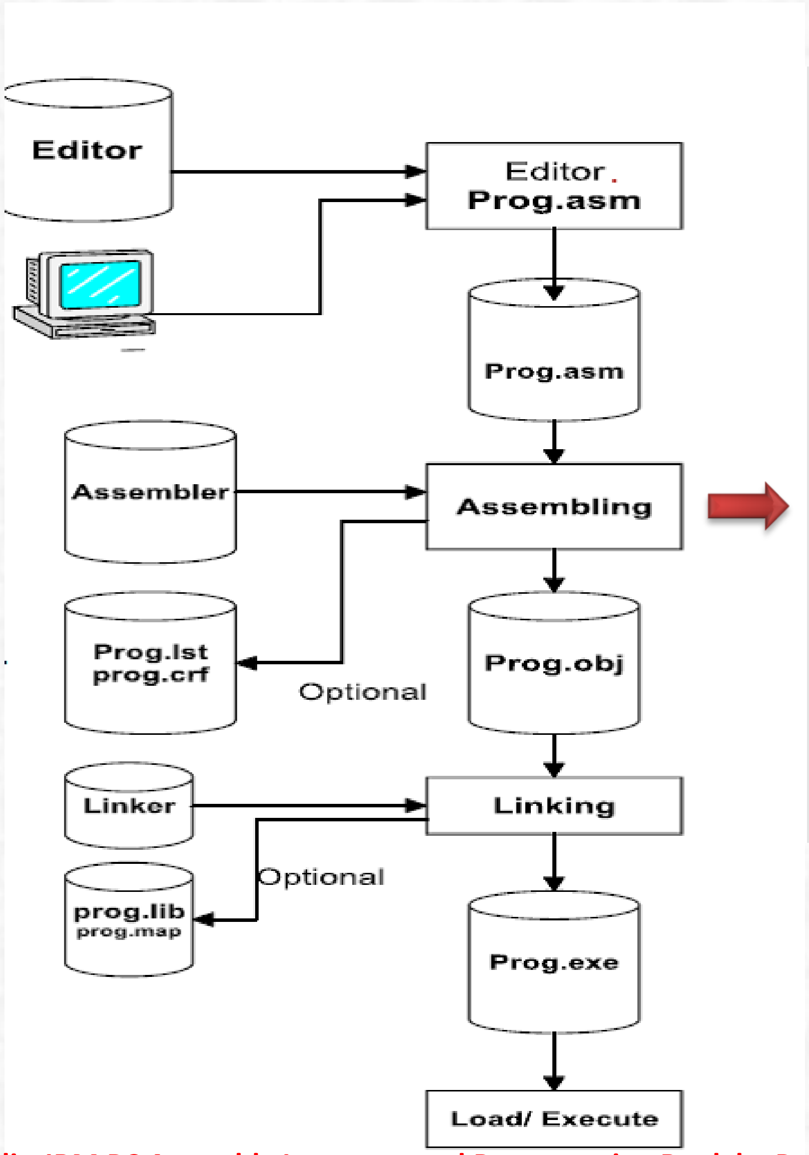
Assembler 1st Pass



```
.MODEL SMALL
.STACK 64
.DATA
X DW 215
Y DW 125
Z DW ?

.CODE

MAIN PROC FAR
L8: MOV AX,0123H ; B82301
MAIN ENDP

END MAIN
```

Image credit: IBM PC Assembly Language and Programming Book by Peter Abel

Assembling, Linking and Executing Programs



Assembler 1st Pass

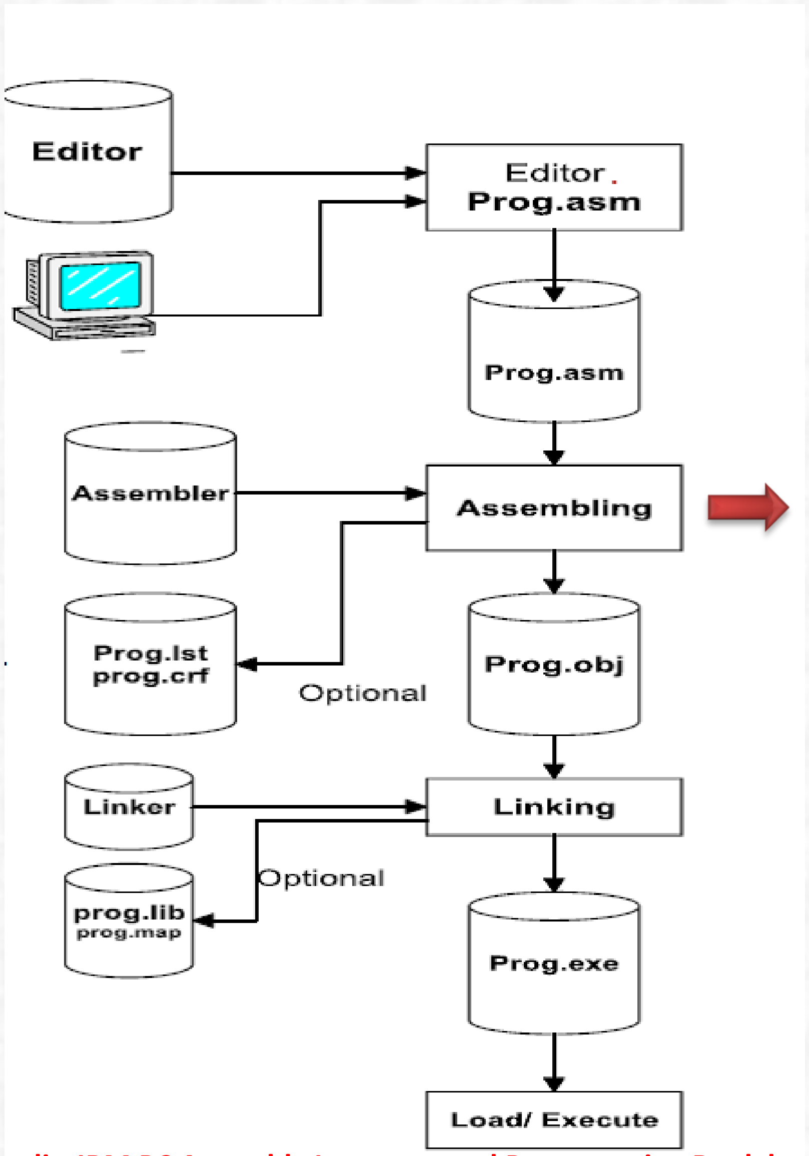
```

.MODEL SMALL
.STACK 64  ---->   00000 - 0003F
.DATA ---------->   00040 - 00045
X DW 215
Y DW 125
Z DW ?

.CODE ------------->   00050 - 00053

MAIN PROC FAR
L8: MOV AX,0123H ; B82301
MAIN ENDP

END MAIN
```

Symbol Table

Symbol	# Bytes	Start	Stop
X	2	00040	00041
Y	2	00042	00043
Z	2	00044	00045
L8	3	00050	00053

Image credit: IBM PC Assembly Language and Programming Book by Peter Abel

Assembling, Linking and Executing Programs

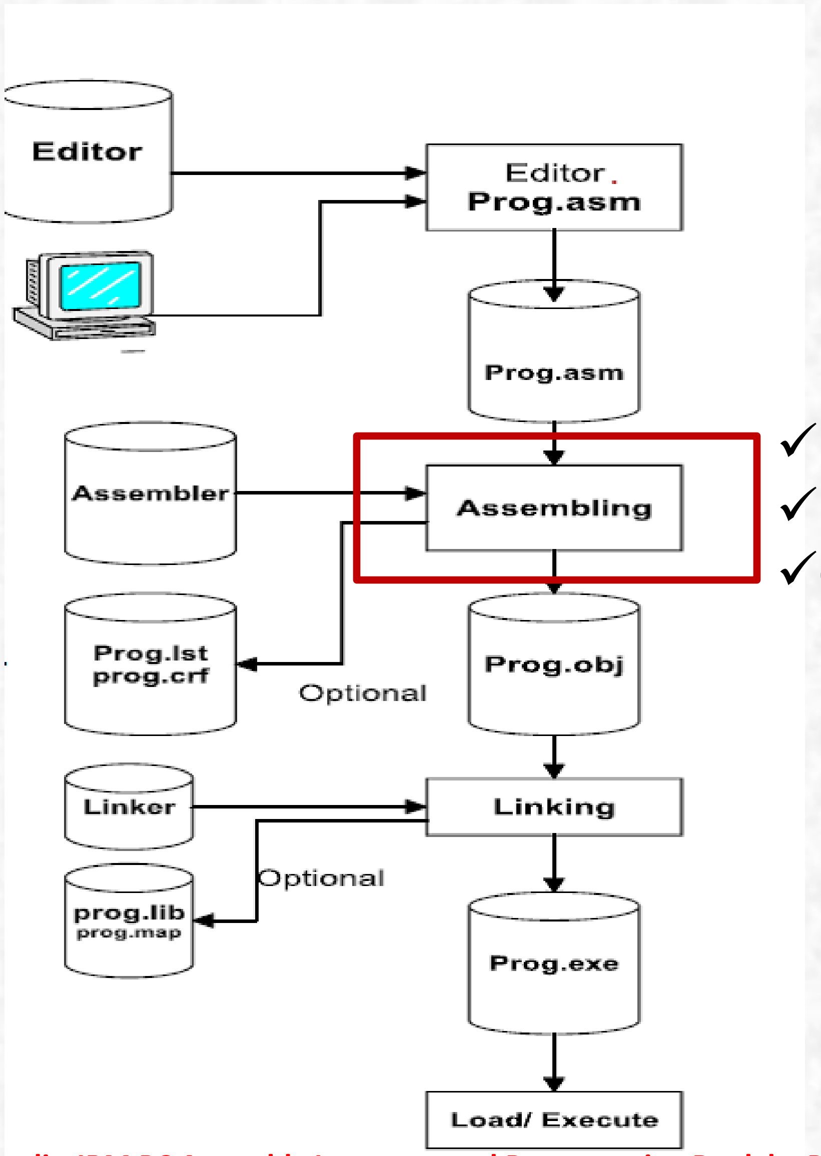


Assembler 2nd Pass

- o Assembler uses the symbol table to generate the complete object code for each instruction.
- oProduce various **.OBJ**, **.LIST**, **.CRF** files.
- o**.OBJ** required for linking a program into **EXE**.
- o**.LIST** for knowing the generated machine code for each instruction, error diagnostic.
- o**.CRF** for which instructions reference which data items.

Image credit: IBM PC Assembly Language and Programming Book by Peter Abel

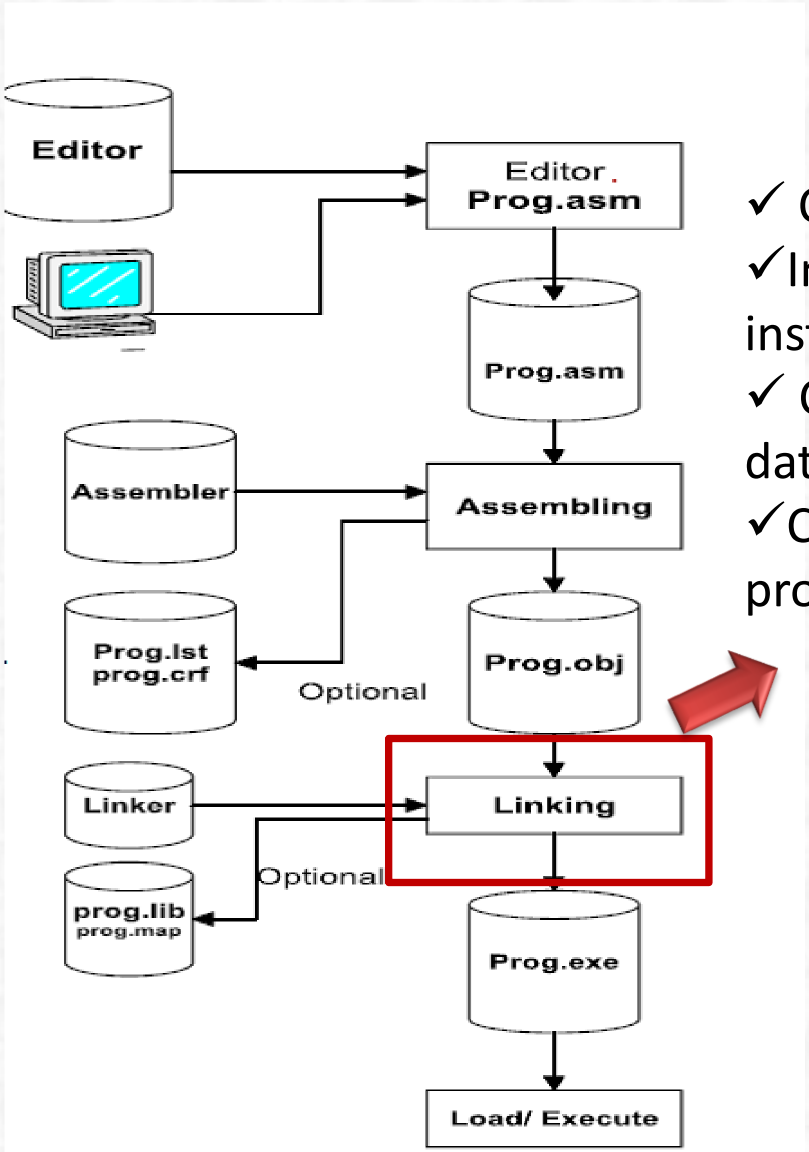
Assembling, Linking and Executing Programs



- ✓ Translate source code into object code.
- ✓ Calculate offsets for data, instruction.
- ✓ Create Header for unresolved offsets.

Image credit: IBM PC Assembly Language and Programming Book by Peter Abel

Assembling, Linking and Executing Programs



✓ Convert **.OBJ** into **.EXE**.

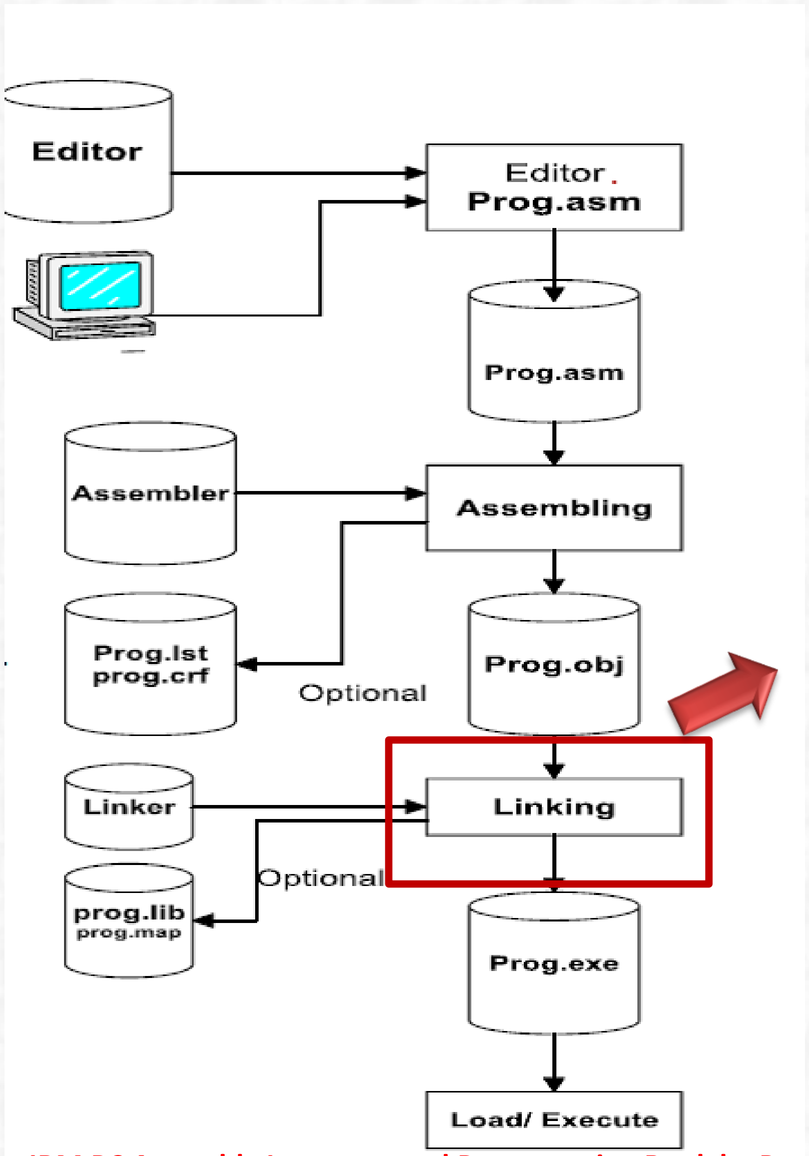
✓Initialize .EXE module with specialized instructions for loading and execution.

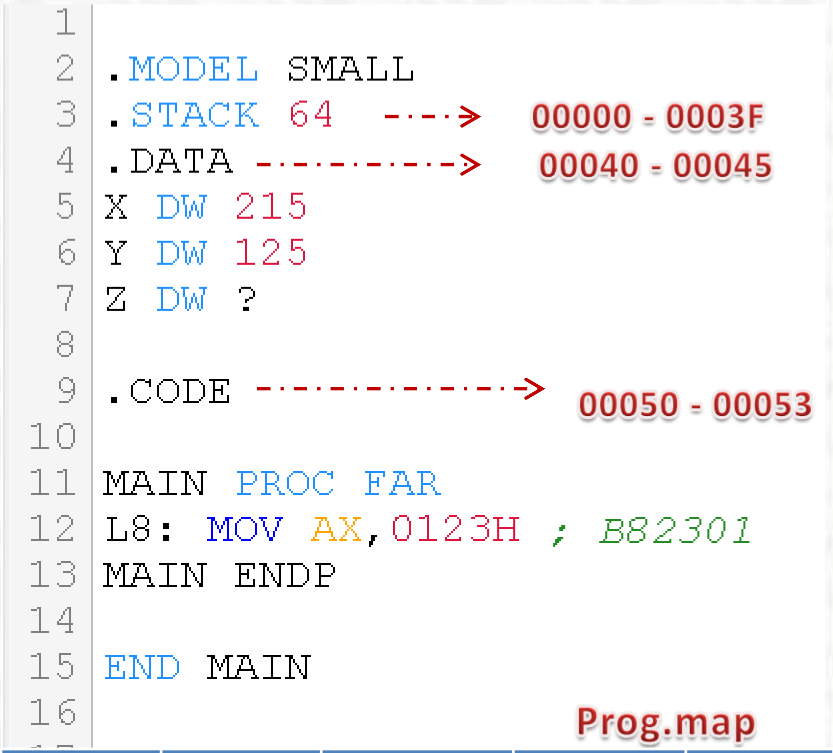
✓ Complete any unresolved offsets for data, instruction.

✓Combine a separately assembled programs into one executable module.

Image credit: IBM PC Assembly Language and Programming Book by Peter Abel

Assembling, Linking and Executing Programs





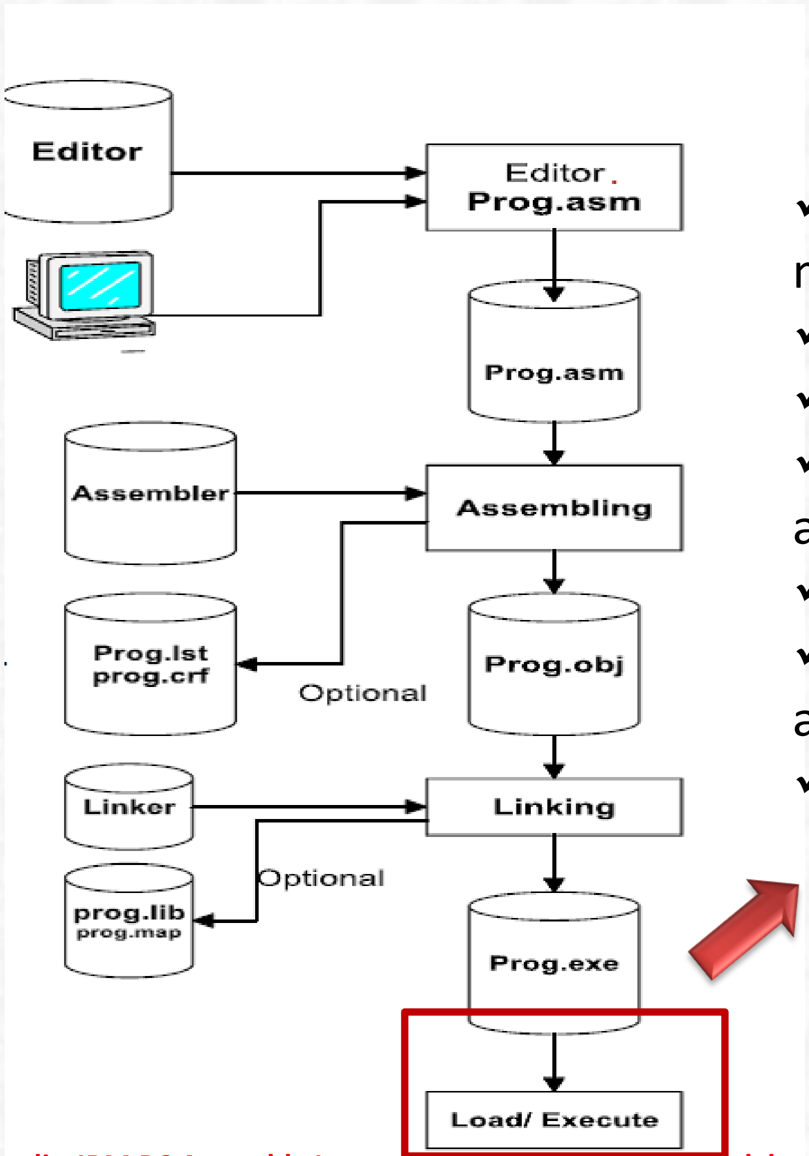
```
.MODEL SMALL
.STACK 64    ---->   00000 - 0003F
.DATA ---------->   00040 - 00045
X DW 215
Y DW 125
Z DW ?

.CODE ----------------> 00050 - 00053

MAIN PROC FAR
L8: MOV AX,0123H ; B82301
MAIN ENDP

END MAIN
```

Prog.map

START	STOP	LENGTH	NAME	CLASS
00050	00053	0003H	_TEXT	CODE
00040	00045	0006H	_DATA	DATA
00000	0003F	0040H	STACK	STACK
Program entry point 0005:0000				

Image credit: IBM PC Assembly Language and Programming Book by Peter Abel

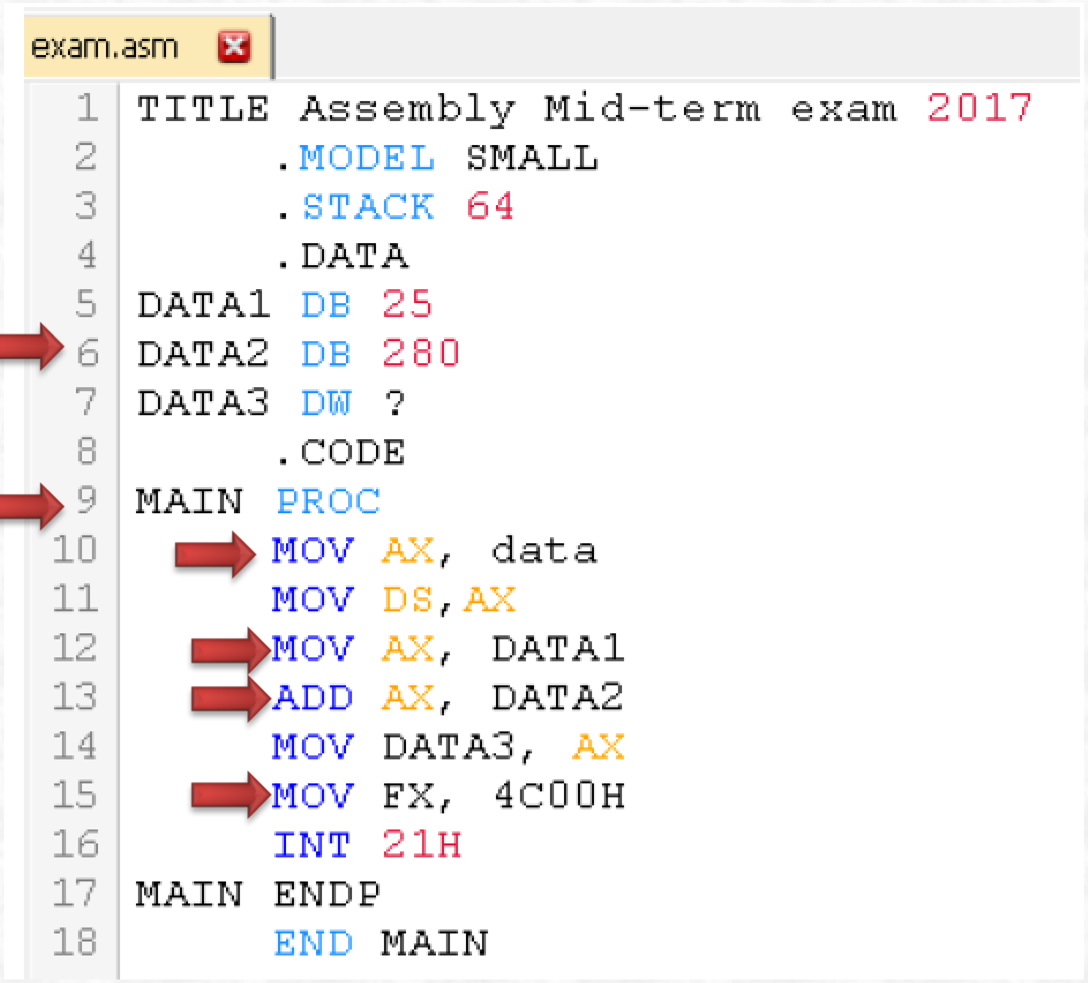
Assembling, Linking and Executing Programs



✓ Load a program for execution in memory.
✓Resolve any incomplete offsets.
✓Drops the header created in .OBJ file
✓Construct 256-byte PSP on the available segment and store .EXE after it.
✓Loads address of PSP in DS, ES.
✓Loads address of Code segment in CS and offset in IP.
✓Initializes SS, SP, etc.

Image credit: IBM PC Assembly Language and Programming Book by Peter Abel

Q



```
TITLE Assembly Mid-term exam 2017
     .MODEL SMALL
     .STACK 64
     .DATA
DATA1 DB 25
DATA2 DB 280
DATA3 DW ?
     .CODE
MAIN PROC
     MOV AX, data
     MOV DS,AX
     MOV AX, DATA1
     ADD AX, DATA2
     MOV DATA3, AX
     MOV FX, 4C00H
     INT 21H
MAIN ENDP
     END MAIN
```