

System Softwares 2

LINKAGE EDITOR LECTURE04 -2023

Dr. Azam E. Al-Rawachy

[COMPANY NAME] | [COMPANY ADDRESS]



1 Introduction

Both **linking loader** and **linkage editor** are tools used in the process of linking object files and creating executable programs.

A **linking loader** is a program that loads an object file into memory and links it with other object files to create an executable program. The process involves resolving external references, relocating the program to the proper memory location, and linking with system libraries. Linking loaders are often used in operating systems to create executable programs from object files.

On the other hand, a **linkage editor** is a program that performs the same tasks as a linking loader, but in a separate step. Object files are linked together by the linkage editor to create a single object file, which is then loaded into memory by a separate loader to create an executable program. The main advantage of using a linkage editor is that it allows for faster program development, as object files can be updated and linked separately without having to link the entire program each time a change is made.

In summary, the key **difference** between a linking loader and a linkage editor is that a linking loader performs linking and loading in a single step, while a linkage editor performs linking in a separate step before the loading process.

Aspect	Linking Loader	Linkage Editor
Function	Loads and links object files to create an executable program in a single step	Links object files in a separate step to create a single object file, which is then loaded by a separate loader to create an executable program
Usage	Typically used in operating systems to create executable programs from object files	Used in software development to speed up the linking process and allow for easier updates to object files
Process	Resolves external references, relocates program to proper memory location, links with system libraries	Resolves external references, relocates program to proper memory location, and links object files to create a single object file
Advantages	Simplifies the program development process by creating an executable program in one step	Allows for faster program development and easier updates to object files
Disadvantages	Slower program development if object files need to be updated frequently	Adds an extra step to the linking process, which may increase the overall development time



Generate the object program for the following two code sections using Two-Pass Assembler and *Linkage Editor*

PRG:

BEGIN:	JMP	MAIN	EB --
N1	DW	-3	
N2	DW	3	
MAIN	MOV	AX, N1	A1 -- --
	CALL	MLL	E8 -- --
	CMP	WORD PTR RES+2, 0	83 3E -- -- --
	JNE	CONT	75 --
	MOV	WORD PTR RES+2,-1	C7 06 -- -- -- --
CONT:	RET		C3
	END		

SUB:

MLL:	MUL	WORD PTR N2	F7 26 -- --
	MOV	RES,AX	A3 -- --
	MOV	RES+2,DX	89 16 -- --
	RET		C3
	END		
RES	DW	0, 0	
	END		



Inserting ENTRY and EXTRN to the PRG and SUB

PRG:

```
ENTRY    N2
EXTRN    MLL, RES
BEGIN:   JMP    MAIN
N1       DW    -3
N2       DW    3
MAIN     MOV    AX, N1
         CALL   MLL
         CMP    WORD PTR RES+2, 0
         JNE    CONT
         MOV    WORD PTR RES+2, -1
CONT:    RET
```

SUB:

```
ENTRY    MLL, RES
EXTRN    N2
MLL:     MUL    WORD PTR N2
         MOV    RES, AX
         MOV    RES+2, DX
         RET
         END
RES      DW    0, 0
         END
```



PRG:

Pass-1

Intermediate file:

Address	Mnemonic field			Error Flag
0000	BEGIN:	JMP	MAIN	0
0002	N1	DW	-3	0
0004	N2	DW	3	0
0006	MAIN:	MOV	AX, N1	0
0009		CALL	MLL	0
000C		CMP	WORD PTR RES+2, 0	0
0011		JNE	CONT	0
0013		MOV	WORD PTR RES+2, -1	0
0019	CONT:	RET		0
001A		END		0

SYMTAB:

Symbol	Value	Error Flag
BEGIN:	0000	0
N1	0002	0
N2	0004	0
MAIN	0006	0
CONT:	0019	0



Pass-2 → Listing file:

Line No.	Address	Mnemonic field			Machin code	Error flag
1	0000	BEGIN:	JMP	MAIN	EB 04	0
2	0002	N1	DW	-3	FD FF	0
3	0004	N2	DW	3	03 00	0
4	0006	MAIN:	MOV	AX, N1	A1 02 00	0
5	0009		CALL	MLL	E8 00 00	0
6	000C		CMP	WORD PTR RES+2, 0	83 3E 00 00 00	0
7	0011		JNE	CONT	75 06	0
8	0013		MOV	WORD PTR RES+2, -1	C7 06 00 00 FF FF	0
9	0019	CONT:	RET		C3	0
10	001A		END			

BEGIN:	0000	EB	
	0001	04	
N1	0002	FD	
	0003	FF	
N2	0004	03	
	0005	00	
MAIN:	0006	A1	
	0007	02	N1
	0008	00	
	0009	E8	
	000A	00	MLL
	000B	00	
	000C	83	
	000D	3E	
	000E	00	RES
	000F	00	
	0010	00	
	0011	75	
	0012	06	
	0013	C7	
	0014	06	
	0015	00	RES+2
	0016	00	
	0017	FF	
	0018	FF	
CONT:	0019	C3	

Object Program:

```

H PRG - - - 0000 001A
T 0000 08 EB 04 FD FF 03 00 A1 02
T 0008 08 00 E8 00 00 83 3E 00 00
T 0010 08 00 75 06 C7 06 00 00 FF
T 0018 02 FF C3
M 0007 + PRG
M 000A + SUB
M 000E + SUB
M 0015 + SUB
D N2- - - - 0004 → For ENTRY
R MLL - - - + SUB → For EXTURNS
R RES - - - + SUB → For EXTURNS
E 0000

```



SUB:

Pass-1

Intermediate file:

Address	Mnemonic field			Error Flag
0000	MLL:	MUL	WORD PTR N2	0
0004		MOV	RES, AX	0
0007		MOV	RES+2, DX	0
000B		RET		0
000C		END		0
000C	RES	DW	0, 0	0
0010		END		0

SYMTAB:

Symbol	Value	Error Flag
MLL:	0000	0
RES	000C	0

Pass-2

Listing file:

Line No.	Address	Mnemonic field			Machin code	Error flag
1	0000	MLL:	MUL	WORD PTR N2	F7 26 00 00	0
2	0004		MOV	RES, AX	A3 0C 00	0
3	0007		MOV	RES+2, DX	89 16 0E 00	0
4	000B		RET		C3	0
5	000C		END			0
6	000C	RES	DW	0, 0	00 00 00 00	0
7	0010		END			0



MLL:	0000	F7	
	0001	26	
	0002	00	N2
	0003	00	
	0004	A3	
	0005	0C	RES
	0006	00	
	0007	89	
	0008	16	
	0009	0E	RES+2
	000A	00	
	000B	C3	
RES	000C	00	
	000D	00	
RES+2	000E	00	
	000F	00	

Object Program:

H SUB - - - 0000 0010
T 0000 08 F7 26 00 00 A3 0C 00 89
T 0008 08 16 0E 00 C3 00 00 00 00
M 0002+ PRG → Modification
M 0005 + SUB → Modification
M 0009 + SUB → Modification
D MLL - - - 0000 → For ENTRY
D RES - - - 000C → For ENTRY
R N2 - - - - → For EXTURN
E 0000

Linking (Linkage Editor) ESTAB (Extended SYMTAB)

Symbol	Value	Error Flag	
PRG	0000	0	0000H + 0000H = 0000H
BEGIN:	0000	0	0000H + 0000H = 0000H
N1	0002	0	0002H + 0000H = 0002H
N2	0004	0	0004H + 0000H = 0004H
MAIN:	0006	0	0006H + 0000H = 0006H
CONT	0019	0	0019H + 0000H = 0019H
SUB	001A	0	0000H + LENGTH(PRG:1A) = 001A H
MLL:	001A	0	0000H + LENGTH(PRG:1A) = 001A H
RES	0026	0	000CH + LENGTH(PRG:1A) = 0026 H



BEGIN:	0000	EB	
	0001	04	
N1	0002	FD	
	0003	FF	
N2	0004	03	
	0005	00	
MAIN:	0006	A1	
	0007	02	N1
	0008	00	
	0009	E8	
	000A	0E	MLL
	000B	00	
	000C	83	
	000D	3E	
	000E	28	RES
	000F	00	
	0010	00	
	0011	75	
	0012	06	
	0013	C7	
	0014	06	
	0015	28	RES+2
	0016	00	
	0017	FF	
	0018	FF	
CONT:	0019	C3	
MLL:	001A	F7	
	001B	26	
	001C	04	N2
	001D	00	
	001E	A3	
	001F	26	RES
	0020	00	
	0021	89	
	0022	16	
	0023	28	RES+2
	0024	00	
	0025	C3	
RES	0026	00	
	0027	00	
RES+2	0028	00	
	0029	00	

Object program:

H Linked 0000 002A

T 0000 08 EB 04 FD FF 03 00 A1 02

T 0008 08 00 E8 0E 00 83 3E 28 00

T 0010 08 00 75 06 C7 06 28 00 FF

T 0018 08 FF C3 F7 26 04 00 A3 26

T 0020 08 00 89 16 28 00 C3 00 00

T 0028 02 00 00

M 0007 + Linked

M 000A + Linked

M 000E + Linked

M 0015 + Linked

M 001C + Linked

M 001F + Linked

M 0023 + Linked

E 0000

Disp = 1AH-09H-3 = 000EH



BEGIN:	C200	EB	
	C201	04	
N1	C202	FD	
	C203	FF	
N2	C204	03	
	C205	00	
MAIN:	C206	A1	
	C207	02	N1
	C208	C2	
	C209	E8	
	C20A	0E	MLL
	C20B	00	
	C20C	83	
	C20D	3E	
	C20E	28	RES
	C20F	C2	
	C210	00	
	C211	75	
	C212	06	
	C213	C7	
	C214	06	
	C215	28	RES+2
	C216	C2	
	C217	FF	
	C218	FF	
CONT:	C219	C3	
MLL:	C21A	F7	
	C21B	26	
	C21C	04	N2
	C21D	C2	
	C21E	A3	
	C21F	26	RES
	C220	C2	
	C221	89	
	C222	16	
	C223	28	RES+2
	C224	C2	
	C225	C3	
RES	C226	00	
	C227	00	
RES+2	C228	00	
	C229	00	

Object program:

- Get PROGADDR from O.S
- SET CSADDR TO PROGADDR
for the first control section assume

PROGADDR = **C200H** “Link and Load SUB + PRG”