INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

System Software
CSN-252
Program Linking & Loading
Ref: 1. L. L. Beck Book
2. R. E. Bryant Chap. On Linkers

Pass 2 Program Logic

- Pass 2:
  - » perform the actual loading, relocation, and linking
- Process Text record and Modification record
- Modification record
  - » lookup the symbol in ESTAB
- End record for a main program
  - » transfer address

```
Pass 2:
begin
  1. set CSADDR to PROGADDR
  2. set EXECADDR to PROGADDR
  while not end of input do
  begin
          read next input record (Header record)
          set CSLTH to control section length
          while record type != E do
          begin
                  read next input record
                  if record type == 'T' then
                  begin
                      a. convert object code to internal representation (if needed)
                      b. move object code from record to location (CSADDR + specified addr)
                  end {if T}
                  else if record type == 'M' then
                  begin
                     search ESTAB for modifying symbol name
                     if found then
                          c. add /subtract symbol value at location (CSADDR + specified addr)
                     else
                             set error flag
                  end {if M}
          end {while}
          3. if (address specified in End record) then set EXECADDR to CSADDR + specified addr)
          4. add CSLTH to CSADDR
  end {while}
  5. jump to location given by EXECADDR
end
```

Records in a CS

H ~~D~~ ~~R~~ T M E

Improve Efficiency

- Use <u>local searching</u> instead of multiple searches of ESTAB for the same symbol

  - assign a reference number to each external symbol

  - the reference number is used in Modification records

- Implementation

  - 01: control section name

  - other: external reference symbols

```
H PROGA 000000 000063
D LISTA 000040 ENDA 000054
R 02LISTB  03ENDB   04LISTC  05ENDC
:
T0000200A 03201D 77100000 050014
:
T0000540F000014 FFFFF6 00003F 000014 FFFFC0
M000024 05 +02
M000054 06 +04
M000057 06 +05
M000057 06 –04
M00005A 06 +05
M00005A 06 –04
M00005A 06 +01
M00005D 06 –03
M00005D 06+02
M000060 06 +02
M000060 06 –01
E000020
```

M000024 05 +LISTB
M000054 06 +LISTC
M000057 06 +ENDC
M000057 06 –LISTC
M00005A 06 +ENDC
M00005A 06 –LISTC
M00005A 06 +PROGA
M00005D 06 –ENDB
M00005D 06+LISTB
M000060 06 +LISTB
M000060 06 –PROGA

Ref No.	Symbol	Address
1	PROGA	4000
2	LISTB	40C3
3	ENDB	40D3
4	LISTC	4112
5	ENDC	4124

PROGA

Avoids multiple searches of ESTAB for the same symbol during the loading of a control section

Ref No.	Symbol	Address
1	PROGB	4063
2	LISTA	4040
3	ENDA	4054
4	LISTC	4112
5	ENDC	4124

PROGB

Ref No.	Symbol	Address
1	PROGC	4063
2	LISTA	4040
3	ENDA	4054
4	LISTB	40C3
5	ENDB	40D3

PROGC

Machine-Independent Loader / Linker Features

Most loaders include

- The use of an **automatic library search** to process for handling external reference

- Some **common options** that can be selected at the time of loading and linking

- Many loaders have a **special command language** that is used to specify options

---

Automatic Library Search

- Most linking loaders can automatically include routines from a subprogram library into the program being loaded

  - Standard library (ex. libc.a)

  - Other libraries (ex. libm.a)

- Programmer only mentions the name (SIC/XE) of these subroutines as external references

- -lm

- -L /usr/lib/…/xyz

- gcc –print-file-name = libc.a

Automatic Library Search

- Linking loaders keeps track of external symbols that are referred to, but not defined, in the primary input to the loader

- Enter symbols in refer record also in ESTAB if not present

- At the end of Pass 1, the symbols in ESTAB that remain undefined represent unresolved external references.

Automatic Library Search

- The loader searches the library or libraries specified for routines that contain the definitions of these symbols, and

- Subroutines fetched from a library in this way may themselves contain external references.

- Repeat the library search process until all references are resolved.

- If unresolved external references remain after the library search is completed, these must be treated as errors.

Loader Options

INCLUDE program-name (library-name)

❑ Directs the loader to read the designated object program from a library and treat it as if it were part of the primary loader input

DELETE csect-name

❑ Allows the user to delete entire control sections

CHANGE name1, name2

❑ Causes the external symbol name1 to be changed to name2 wherever it appears in the object programs.

```
INCLUDE    READ(UTLIB)
INCLUDE    WRITE(UTLIB)
DELETE     RDREC, WRREC
CHANGE     RDREC, READ
CHANGE     WRREC, WRITE
```

Loader Options

LIBRARY MYLIB

❑ Involves the automatic inclusion of library routines to satisfy external references

❑ Such user-specified libraries are normally searched before the standard system libraries.

❑ This allows the user to use special versions of the standard routines.

NOCALL STDDEV, PLOT, CORREL

❑ Instructs the loader that these external references are to remain unresolved.

❑ Avoids the overhead of loading and linking the unneeded routines, and saves the memory space.

Linking loader

o performs all linking and relocation operations, including automatic library search if specified, and loads the linked program directly into memory for execution.

Linkage editor

o perform linking prior to load time

o produces a linked version of the program (load module or executable image), which is written to a file or library for later execution.

Dynamic linking

o linking function is performed at execution time.