Today’s Topics

• Beck’s Section 3.2.3: Algorithm and Data Structure for a Linking Loader
Algorithm and Data Structure

- Two Passes Logic
  - Pass 1: Assign addresses to all external symbols
  - Pass 2: Perform the actual loading, relocation, and linking
- ESTAB (external symbol table)

<table>
<thead>
<tr>
<th>Control section</th>
<th>Symbol</th>
<th>Address</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program A</td>
<td>LISTA</td>
<td>4000</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>ENDA</td>
<td>4054</td>
<td></td>
</tr>
<tr>
<td>Program B</td>
<td>LISTB</td>
<td>4063</td>
<td>7F</td>
</tr>
<tr>
<td></td>
<td>ENDB</td>
<td>41D3</td>
<td></td>
</tr>
<tr>
<td>Program C</td>
<td>LISTC</td>
<td>4112</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>ENDC</td>
<td>4124</td>
<td></td>
</tr>
</tbody>
</table>

Pass 1 Program Logic

- Pass 1:
  - Assign addresses to all external symbols
- Variables & Data structures
  - PROGADDR (program load address) from OS
  - CSADDR (control section address)
  - CSLTH (control section length)
  - ESTAB
- Fig. 3.11(a)
  - Process Define Record
Pass 2 Program Logic

- **Pass 2:**
  - Perform the actual loading, relocation, and linking
- **Modification record**
  - Lookup the symbol in ESTAB
- **End record for a main program**
  - Transfer address
- **Fig. 3.11(b)**
  - Process Text record and Modification record
Fig. 3.11(b)

Pass 2:

begin
set CRGDR to PROGRDR
set EXECOND to PROGRDR
while not end of input do
begin
read next input record (Header record)
set CEUM to control section length
while record type ≠ 'B' do
begin
read next input record
if record type = 'Y' then
begin
(if object code is in character form, convert
into internal representation)
move object code from record to location
(CSAADDR + specified address)
end (if 'Y')
else if record type = 'H' then
begin
search ESTAB for modifying symbol name
if found then
add or subtract symbol value at location
(CSAADDR + specified address)
else
set error flag (undefined external symbol)
end (if 'H')
end (while ≠ 'E')
if an address is specified (in END record) then
set EXECOND to (CSADDR + specified address)
add CEUM to CRGDR
end (while not END)
jump to location given by EXEOND (to start execution of loaded program)
end (Pass 2)

Figure 3.11(b)  Algorithm for Pass 2 of a linking loader.

Improve Efficiency

- Use local searching 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

- Example
  - Fig. 3.12
### Fig. 3.12 (1/3)

<table>
<thead>
<tr>
<th>Ref No.</th>
<th>Symbol</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PROGA</td>
<td>4000</td>
</tr>
<tr>
<td>2</td>
<td>LISTB</td>
<td>40C3</td>
</tr>
<tr>
<td>3</td>
<td>ENDB</td>
<td>40D3</td>
</tr>
<tr>
<td>4</td>
<td>LISTC</td>
<td>4112</td>
</tr>
<tr>
<td>5</td>
<td>ENDC</td>
<td>4124</td>
</tr>
</tbody>
</table>

Fig. 3.12 Object programs corresponding to Fig. 3.8 using reference numbers for code modification. (Reference numbers are underlined for easier reading.)

---

### Fig. 3.12 (2/3)

<table>
<thead>
<tr>
<th>Ref No.</th>
<th>Symbol</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PROGB</td>
<td>4063</td>
</tr>
<tr>
<td>2</td>
<td>LISTA</td>
<td>4040</td>
</tr>
<tr>
<td>3</td>
<td>ENDA</td>
<td>4054</td>
</tr>
<tr>
<td>4</td>
<td>LISTC</td>
<td>4112</td>
</tr>
<tr>
<td>5</td>
<td>ENDC</td>
<td>4124</td>
</tr>
</tbody>
</table>
Fig. 3.12 (3/3)

<table>
<thead>
<tr>
<th>Ref No.</th>
<th>Symbol</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PROGC</td>
<td>4063</td>
</tr>
<tr>
<td>2</td>
<td>LISTA</td>
<td>4040</td>
</tr>
<tr>
<td>3</td>
<td>ENDA</td>
<td>4054</td>
</tr>
<tr>
<td>4</td>
<td>LISTB</td>
<td>40C3</td>
</tr>
<tr>
<td>5</td>
<td>ENDB</td>
<td>40D3</td>
</tr>
</tbody>
</table>

Figure 3.12 (cont’d)