Loaders and Linkers (continued)

External Symbols in C:

- Refer to Handout 7.1.
- Compiler must also handle the `static` attribute for functions and variables.

A Representation for EDT and ERT:

- EDT and ERT as part of object code itself.
- Definition Record (D-record) for EDT: Each D-record has a symbol and its relative address.
- Reference Record (R-record) for ERT:
  - R-record contains all the external symbols referenced by the module.
  - Addresses will appear in modifier records.
  - R-records strictly unnecessary; useful in making the linking process more efficient.

Changes to Modifier Records:

- Current format: Each modifier record contains starting byte address and number of bytes.
- Adequate for relocation but not for linking.
- New format: Each modifier record contains starting byte address, number of bytes, a flag and a local or external symbol.
- Flag can be '+' or '-'.
- New format can handle both relocation and linking.

Example: To be presented in class. (The example is based on Handout 6.2.)

Local and External Symbols:

- **Example:** To be presented in class.
- Note the difference in a modifier record when an external symbol defined in a module is also referenced in the same module.

Need for the '-' Flag:

- Operand field may have an expression containing external symbols.
- These expressions are generally simple. (They contain only '+' and '-' operators).

Example:

```
FUNCT    CSECT
         EXTREF   FIRST,LAST
           .
           .
SIZE     WORD     LAST-FIRST+1
```

- Suppose the LC value of SIZE is 70.
- Assembler initializes bytes 70, 71 and 72 to 0, 0 and 1 respectively.
- Assembler writes the following modifier records to the object object file.

```
M   70   3   +  LAST
M   70   3   -  FIRST
```


Algorithm for Linking Loader:

- **Input:** Object code for each module organized as Header record, D-record, R-record, Text records and End record.
- **Assumptions:**
  - Only one module specifies a starting address in its End record.
  - Modules loaded successively with no gaps.
- Algorithm uses two passes.
- This version of algorithm ignores R-records.
- **Pass I:** Builds the External Symbol Table by combining the EDTs of the modules.
- **Pass II:** Resolves the external symbols, performs relocation and loading and starts execution.
- **Algorithm outline:** Handout 7.2.
- **Reading assignment:** Figures 3.11(a) and 3.11(b) of [Beck].

Usefulness of R-records:

- With the revised format for modifier records, R-records are strictly unnecessary.
- R-records can be used to improve the efficiency of the linking process.

Example: To be discussed in class.