CSI 402 – Lecture 7

(Final Remarks on Assemblers)

Program in Multiple Files (Control Sections)

- Modules that can be separately assembled; usually placed in separate files.
- Executable version of the program created by “linking” together all the separately assembled modules.
- New assembler directive: `CSECT` (for “control section”).

Example:

```
# Main program -- also a control section.

MAIN    START   0     #Main program
          .
          .
        +JSUB   SORT
        +JSUB   MAX
          .
          .
```

Control Sections (continued)

Example (continued):

```
SORT      CSECT          #Subroutine.
            .
            .
            .
          RSUB

MAX      CSECT          #Subroutine.
           .
           .
           .
         RSUB
```

Note: We may have each of the above subroutines in a different file.

Modifications to the Assembler

Note: Assume that the source program is in two files: one containing `MAIN` and the other containing `SORT` and `MAX`.

- Assembler needs to know that `SORT` and `MAX` are in a different file.
- New assembler directive: `EXTREF`.

Example:

```
        EXTREF   SORT, MAX
```

- The `EXTREF` directive will be used in the file containing `MAIN`.

Modifications to the Assembler (continued)

- SIC/XE statements referencing external symbols must use the 4-byte format.
- Assembler cannot fill the address part of such instructions. (It will be done by the linker.)
- Assembler must produce an **External Reference Table** (ERT).
- Each entry of ERT has a symbol and the relative address where the address of the symbol is needed.
- Each control section may have an ERT.

Example: See Handout 7.1.

Modifications to the Assembler (continued)

- Another new assembler directive: `EXTDEF`.

Example:

```
        EXTDEF    SORT, MAX
```

- The `EXTDEF` directive will be used in the file containing `SORT` and `MAX`.
- Assembler must produce an **External Definition Table** (EDT).
- Each entry of EDT has a symbol and the relative address where the symbol is defined.
- Each control section may have an EDT.

Example: See Handout 7.1.

Modifications to the Assembler (continued)

Differences between ERT and EDT:

- A symbol may appear two or more times in an ERT. No symbol can appear more than once in an EDT.
- A symbol may appear in two or more ERTs. No symbol may appear in more than one EDT.

Algorithms for creating ERT and EDT:

- See Handout 7.2.

Modifications to the Assembler (continued)

List of Tables used by an Assembler:

- Symbol Table (ST)
- Machine Opcode Table (MOT)
- External Reference Table (ERT)
- External Definition Table (EDT)

Suggested Exercises:

- Make sure that you understand the need for and the differences between EDT and ERT.
- Create small examples of SIC/XE programs and practice constructing ERTs and EDTs.
- **Reading Assignment:** Section 2.5 of [Beck]. (Contains a discussion of some commercial assemblers.)