CS530, Spring 2021, Project #2

28 Feb 2021

You and your team shall develop, test, and deliver a link-editor program for the XE variant of the SIC/XE family of machines.

XE LINK-EDITOR REQUIREMENTS:
The XE link-editor program shall open SIC/XE assembler listing files (refer to fig 2.3 in your text) and generate executable object file(s) for the XE machine and the ESTAB for the project (note, do not build a link-loader!).
Input - The user will start the program (the program shall be named "led") and will provide the listing file(s) as arguments on the command-line each separated by spaces, i.e.:

"[cssc0000@edoras ~]$ led first.sl second.sl third.sl"

Note, if no arguments are provided on the command line, your program shall print a friendly message stating why you are stopping and then terminate the program.

The specification format of the input file is SIC/XE Assembler Listing Format, this is similar to the listing file found in figure 3.8 in the text. You will scan this file and run a check on memory mapping: check that all format 3 & 4 instructions are making memory references within the scope of the program's memory space, if any are out of bounds then print a friendly message stating the issue causing you to stop, then terminate the program. You may assume that there are no other issues with the code (no other errors).

Output – The SIC (XE variant) object file(s) such as those found in figure 3.9 of the text. Print the ESTAB in a separate file (*name.st*) and is such as the ESTAB at the top of page 143 in the text.

TEAMS:
You shall work in teams of two - three people on this project. You may choose to use pair programming, dividing work up, or other methods for work completion, that is up to you although I strongly encourage you to attempt pair programming!

ADDITIONAL REQUIREMENTS:
README file - you shall create a README file; consult the instructions for README file content on the course Blackboard. Also, your source files SHALL CONTAIN sufficient comments for making the source easy to read. Points will be taken off for poorly (or non) commented source or inadequate README file documentation.

Compiler and make (and Makefile) – You shall use C/C++ (gcc/g++) and use make to compile your program for this assignment; you will need to create a Makefile for your project, consult the example Makefile(s) in the course Canvas Resources module. Name the executable, 'led' (Link Editor for the SIC/XE machine).

Test files – You shall prepare and include test files used during the development and test of your project.

Software Design Document - You are required to perform software design of this system. Include a Software Design Document (SDD) and turn it in with your project. Note, you will not be held to formal design specification/formatting or use any of the formal methods. Turn in a file which contains your software design. You may include a kanban (and stories), models, drawings, descriptions, diagrams or similar

tools you used for your system/software design. This is a significant part of your grade and needs to be adequately captured in your documentation. Include a description of how your team was organized and how effectively you worked together and areas for improvement.

Make sure that all files (README, source files, header files, Makefile) contains each team member's names and RedIDs!

TURNING IN YOUR WORK:

The project (assignment #2) is due at 1700, Wednesday, 14 April 2021

- **A draft copy of your SDD is due at 1700, Monday, 15 March 2021**

Your project shall include C/C++ source files, an include/header file, a Makefile, and a README file. ONLY ONE MEMBER OF YOUR TEAM TURNS IN THE PROJECT. To turn in your project, each team shall select one person, all files shall be in that person's class account on edoras in a directory named "a2" (~/a2). Leave any test files in this directory as well. BE SURE ALL TEAM MEMBERS NAMES AND CLASS ACCOUNTS ARE IN THE README FILE. Finally, the designated person turns in the project by uploading a tarball/zip-file with all project files to Blackboard and entering any comments in the assignment's turnin.