Ethan Fetsko

Sarah Rozow

Chandler Henson

Status

We have made pretty substantial progress so far. At the moment, our assembler can do the first pass of the “basic.txt” assembly file completely with no errors. It can also almost completely handle the “control\_sections.txt” file. Our compiler mostly works on the control sections file, but it still need to be able to handle “EQU” statements and it needs to treat each control section as a separate compile. Because it currently is not treating the control sections as separate compiles, it currently has each control section using the same SYMTAB. We will likely make it so that the compiler first runs pass 1 and pass 2 on the first control section, before having it run pass 1 on the second control section.

Project Structure

We decided to go with Java to make our assembler. The reasoning is that assemblers typically use Hash Table data structures to handle the SYMTAB and OPTAB. Java has hash tables defined, so it’s a big time saver to use the predefined hash tables instead of creating our own hash table system in C or C++ (as neither of these languages have hash tables predefined).

As for our project structure, we are using two main classes at the moment to construct our assembler. The first is a “Tables.java” class that simply holds the static OPTAB hash table and the static SYMTAB table. The static SYMTAB could be a problem when doing control sections for the 2nd pass though. If the SYMTAB is static instead of being linked to an object, then we may have trouble passing the SYMTAB data from several different control sections to pass 2. However, we should be able to overcome this though my doing pass 1 and pass 2 of each control section one at a time.

In our assembler class, we have functions to open, close, and read the external files. We also have similar functions to create and close our intermediate file. Lastly, we have some functions to make the “passOne” function more readable. These functions do various things such as converting the addresses from decimal to hexadecimal, a function to increment the LOCCTR based on what opcode is used, and a function that writes out each line to the intermediate file.

Work Division

**Ethan Fetsko** – Primary programming. In charge of implementing the initial assembler features.

**Sarah Rozow** –

**Chandler Henson** –

**Group work** – We have all worked on discussing and designing our approach to the assembler structure. We also all contributed to laying out our schedule for the project.

Future Plan/Milestones