Justin Deines and Meagan Callahan

April 11, 2025

Hack Assembler (HW5)

Overview

Installation and Usage: To make the program usable, place the files CodeParser.java, HackAssembler.java, HashCreator.java, comp.txt, dest.txt, jump.txt and symbols.txt in the same directory. Compile the program with

```
javac HackAssembler.java
```

and run with

```
java HackAssembler file.asm
```

where file.asm is the assembly file you wish to convert to machine language. The resulting machine language with be placed in the same directory, as file.hack.

Description: This program is an Assembler made to convert a Hack assembly code file (ending in .asm) to a 16-bit binary machine code file (ending in .hack) for use in the virtual computer given by nand2tetris. It features a two-pass assembler, allowing for symbol resolution and code cleaning as well as translation. It handles labels, variables, predefined symbols, and C-insctructions (those that contain operations). It is also able to dynamically allocate variables, allowing users to define their own variables in assembly code.

HackAssembler.java: This is the main file for the program, and the one that the user interacts with in order to use it. It calls the other two classes, using HashCreator to manage translation of code, and CodeParser to parse the code to be translated.

CodeParser.java: This is the parser class, which reads a line of code, removes comments and white space, then holds the instruction compenents for use by HackAssembler and HashCreator.

HashCreator.java: This is the hash table class, which holds all the symbols used in assembly and their corresponding machine language translations. It gets the symbols from the files symbols.txt, jump.txt, dest.txt, and comp.txt.

Workload

Meagan Callahan – Created CodeParser and HashCreator classes, as well as symbols.txt, jump.txt, dest.txt, and comp.txt

Justin Deines – Created HackAssembler as well as doing minor bugfixing throughout other files, and wrote the documentation.