Steve Werdick

Michael Mandel

System Documentation

**Preface**

Our hardware emulator is a java project consisting of 5 classes spread across 5 .java files and contained in a single package. This documentation will discuss the purpose and function of each of these classes one at a time.

**Main.java**

This is the class containing our ‘public static void main (String[] args)’ method and where our program begins. In the main method we simply instantiate a few objects, namely the ‘Computer’ and ‘Program’ objects we’ll need to run our programs. Main also contains a while loop which allows us to point to as many different input files as we want during a single execution. The class also is also responsible for reading in the program .txt files which it does via its ‘readFile’ method.

**Program.java**

Program.java is a simple little container class which organizes a few different lists, ints and Strings into a single object representing a program to be run on the emulator. It has two separate lists, one for the data portion of the program and one for the instruction portion. It also has an int for ‘timeLimit’ contained in program headers. The methods are basic getter and setter methods with the exception of ‘getNextDataLine’ which treats the list ‘data’ like a queue and pops the next instruction off of it.

**ProgramParser.java**

This class contains several static methods and is responsible for taking the text file read by Main, and populating a container class called ‘Program’. It contains three methods; ‘parse’ which divides separates a program into its instruction portion and its data portion, and ‘parseData’ and ‘parseInstructions’ which populate their respective parts of a ‘Program’ object which is eventually returned to the ‘Main’ class.

**Computer.java**

This class represents our computers hardware and is responsible for executing programs passed to it. It contains a few Strings, Booleans, and ints representing a toggle, accumulator, instruction register, and instruction counter. It also has a MainMemory object representing, well, the computer’s main memory. It has two methods; ‘runProgram’ which iterates over a programs instructions, and executeInstruction which contains the code responsible for executing each specific instruction contained in a program’s .txt file.

**MainMemory.java**

This class represents our emulators main memory as a two dimensional array of Strings. Much like ‘Program’, it is essentially a container for the String array class with a few complicated getter and setter methods that allow the ‘Computer’ object to get and set specific blocks and words in memory based.