Bushra Hameed

**Project 1**

**Purpose:**

The purpose of this project was to create a simulation of a simple computer that had two major components: CPU and Memory. These two major components were created in two separate processes where they were able to communicate. This project has been coded in Java.

**Implementation:**

The CPU and Memory are able to communicate with one another as the Runtime exec method is used to spawn a child process. The CPU initiates communication with Memory, and the I/O stream acts like pipes. The CPU has a list of registers; PC, SP, IR, AC, X, & Y. The registers are utilized within the instruction sets. The instructions are fetched into the IR from memory, decoded, and executed before the next instruction can be fetched. The Memory consists of two operations: read and write. The read function returns the value at the address, and the write function writes the data to the address. There is also a timer for interrupts that occurs after every X number of functions. The interrupts are processed when the CPU is in the kernel mode and later returns the value from the interrupt and also prevents any nested interrupts. A list of functions were created and utilized within both CPU.java and Memory.java to help tasks to run separately rather than all together. In CPU.java, for example, separate functions were created to read data from a memory address and to send a file name to a memory address. In addition to this, child processes were also able to process data addresses through their own functions as variables were passed by value. These are only a few examples. In Memory, there are separate functions to read the value at an address and write data to the address. In addition to this, the file is read and special characters can be skipped.

**Personal Experience:**

As I worked through this project, I had a better understanding of the material covered in class in regards to how the CPU fetches and executes instructions while also dealing with interrupts. I learned a lot more about Java’s library instructions, more specifically the Runtime exec method. I also learned about split() function and how it could be implemented in other projects. Before using split(), I had attempted using try catch blocks with a series of if-else statements, which was quite tedious. Learning about this function and how to implement it was really interesting. Overall, I learned a lot about operating systems as well as a few new Java methods.