Alan Padilla Chua

Axp141330

Project 1 Summary

The project purpose can be stated as meant to help us understand more the fetch and execute cycle that a processor and memory must undergo in order to run a program. The processor we wrote understands a somewhat assembly like instructions that must be read from a different program which is the child process of the parent program being the memory.

I focused my project in 2 different parts. Because I wrote the project in java I would need 2 separate programs that would need to talk through the streams. For memory it was straight forward. I would read the file name to load into memory as an argument. The first thing memory does is parse the file and write to the main memory array at the correct index as directed by the file. It was very important to correctly parse the lines of the file and handle cases of empty lines and .### number cases. Once the array has been correctly written to, memory waits for instruction by the processor.

The processor was definitely the hardest part to debug and implement. The first step is to correctly use a runtime exec to call the Memory program from the processor. Then set up the communication streams such that all the output that left memory would be in the input stream of the parent. Then I set up my processor to send a string that began with “R” to indicate to memory that I wanted it to send me the value at the PC that I sent. To write to memory an instruction would have to begin with anything other than “R”. For continuity I wrote those commands using the “”W” followed by 2 integers indicating the parameters on where to write to memory. Once a command was sent to memory I would hold the reply in IR. The IR would then fall into a switch statement that would dictate which instruction to execute. Instruction 50 called a the end() method in java that terminated both programs at the same time.

Personally I really enjoyed this project and I learned a lot. It was definitely challenging and required some long hours of debugging the stack push and pop methods and the interrupts. I will definitely start projects for this course sooner rather than later.