# **Summary**

This project was to explore how processes communicate with each other in a generalized CPU/Memory environment and how exactly the environment worked. ranging from the basics of State Saving between modes and how that played with interrupt handling,

to merely understanding how registers interacted with different Instruction calls.

This project was implemented through the use of Java and its libraries, the way I did it was to first make the program without inter-process communication, as that tends to make things impossible to debug. You can see the program I made in the file Project1.java. then after the program was working I separated CPU and Memory functions into different classes and implemented the BufferReader/Writer classes to communicate through processes. instead of copying and pasting busy wait code every time I wished to read/write to the memory I instead created the helper methods readMemory() and writeMemory() which handled all the minutia of communication between processes and helped keep the shape of my original code roughly the same.

As for my personal experience with this project. It was relatively pleasant, had a bunch of fun really trying to figure out why my interrupts weren't working, then realized that I hadn't followed the requirements.

I had a moment of despair after I translated everything from my Project1.java file into the CPU.java and Memory.java classes, but it was quickly solved after I realized I had made the mistake of not initializing my Memory array, that was a huge relief.