Matthew McMillian

CS4348.501

CS4348 – Operating Systems Concepts Project Summary

This project’s purpose was to simulate the interaction between a CPU and MEMORY through multiple (separate) processes by using the *fork()* method in *C*. I constructed my project in C/C++ and used multiple processes that communicated through two pipes (one for reading and one for writing). The CPU contains different variables for determining things such as kernel mode, scheduler, and different registers. The CPU also contains an array of function pointers that simulates the instruction set. The MEMORY contains an array that simulates the memory system. Both parts of the system contain read / write helper functions. My experience with the project was OK. I began by creating the instruction set functions and setting up the initial variables. I decided to make multiple classes instead of doing the project in one file since I felt it was more streamlined and easier to work with. These are the files that hold the variables and objects needed for the system to perform correctly. During the middle of the project I was overthinking the use of the stack and stack pointer and I spent a lot of time trying to research how this part of the project worked. After talking to some other students and talking to Dr. Ozbirn I realized that I was overthinking this part and was able to fix the problems I as having and moved on to the scheduler and syscall section. This section took me a little bit as I had to rework my instruction set’s logic. Overall the project went well; the logic and general objectives were very straightforward, and the instruction set logic was not very difficult. I think that creating multiple classes and using an array of function pointers was a more unique approach to the project that definitely added hardships (such as *typedef* syntax issues within classes), but it made the overall experience of creating the project more eventful and interesting.