**Shiva Modugu**

**sxm220181**

**CS4348.004**

**Project 1: Exploring Multiple Processes and IPC**

**Project Purpose:**

This project provided a deeper understanding of topics covered in the course, particularly forking processes, and communication through pipes, which clarified the concept of processes. The project explored fundamental systems within a processor and allowed for the creation of interesting applications.

**Project Implementation:**

The Java code presents an implementation of a simulated computer system with a CPU and memory interaction. The Project1 java source file is the principal component, which manages the execution of an external file named “Memory.” Memory reads an input program file path, then initializes memory, and then performs input/output operations. The program supports commands for writing and reading files from specific memory addresses. The program will manage file input validation and memory initialization from the input file.

In the Project1 class, I have implemented a nested class called ComputerCPU, which is aimed to represent the CPU of a simulated system. This is a nested class that represents the CPU of the simulated computer system. It has instance variables to store the CPU state and methods to communicate with the memory, fetch and execute instructions, and oversee system calls and interrupts. The computer will use the readMemoryData and writeMemoryData methods to send and receive data from the external process, which is the Memory class, and check for invalid memory access in user mode. The fetch and execute methods are the next instruction from the memory. They just manage the timer interrupt and the end of an execution. I also have a class that runs all the cases that were in the instructions. I also implemented case9 and case30 methods outside of the case files. Given the complexity of the process and the length of the code, implementing these methods outside of the cases allows for ease of understanding for the user. Finally, we have the kernel Mode which is the method that switches CPU to kernel mode and saves the CPU state to the stack. Overall, the Java code provides a comprehensive implementation of a simulator computer system and memory interaction.

**Personal Experience:**

I personally had trouble with this project. I wrote my code in Java as stated before and I could not find a way to rewrite the code into one Java source file. I did not think it was possible given that we needed to use exec and we were only allowed to use processes. The function exec is to execute a specific command and arguments in a separate process with the specified environment, and I could not find way for it to be used to call a class within the same source file. Overall, I believe this project truly has helped me understand processes and how they CPU interacts with the memory.