Kumaravinash Konduru

Kxk151330

CS 4348.501

Project 1 Summary

The purpose of this project was to be able to learn how a computer’s CPU and Memory operate. In more detail though the project was about implementing a program that simulated how a CPU and Main Memory would perform, both with their individual functions and how they would talk to each other. To do this we had to create two classes, one to represent the CPU and one to represent the memory. And to test whether or not our functions worked we had to test them against a set of sample test cases to see whether or not they would provide the correct outcomes. In this project the class that had to represent the memory of the compte had to do certain things. This included reading the inputted file and loading all of the instructions onto a stack to be processed. It also had to split that stack into user and kernel portions depending on what mode the CPU was in. Also the main memory had to be able to read the inputs from the CPU and from that determine whether or not to read an input, write an output, or change user states. When it came to the CPU class it had to be able to read the instruction set from the main memory by using a process and then do a specific action depending on what instruction set it was given. Registers would be need to store and manipulate information within the CPU and were used in different ways depending of what the instruction was. The CPU class also need to the the ability to read and write to the main memory using the process between the two, which is how the CPU was able to get access to the stack. Also the CPU class had to have the ability of have interrupts, meaning when called the CPU would switch to the kernel mode to handle the interrupt to which the previous information would be saved to a stack. After the interrupt the registers were restored and the process starting where it had left off. These interrupts could be initiated by either the program running or by a timer set by the user. All in all the project was intended for people to know and understand how CPU and Memory work by using a high level language to show in a simplistic fashion how it is done.

When it comes to how I implemented this program I’ll start with the Memory class first. The first thing that I did for this class was to get fstream inputs work, this was so that I could get my program to output the contents of a sample file line by line. After this I narrowed it down so that it would only output the first integer, if there was any at all. After this I then saved all of the instructions onto any integer array that I used to represent my stack. After this I wrote my read and write functions to the stack that would be used whenever the CPU would send an instruction. Next I worked and the Scanner and main function. Here the Scanner input would receive an output from the CPU which was basically a command either being to read in the data at a particular address, write a piece of data to a particular address, end the program, and enter or exit the kernel mode. With all of this my Memory class was basically complete and after that was done I went on and started to program the CPU class. This was a much harder class to program as it was the one that did most of the logic of the program and as such had many more moving parts to it. The first this that I did to implement this class was to create the process that I would be using in order to communicate with both of the programs. Because I was programming in Java I was able to create a process by calling the Runtime exec function. This function allowed for me to run the Memory class as well as create an inputstream process and output stream process between the two programs. After I had created the runtime process and was able to feed in the instructions into the CPU I then went on to create the instruction set. Here I first initialized the registers I was going to be using as global variables and then created an instruction set function that would take in an instruction as an argument and then use a switch statement to find what action or set of actions that instruction corresponded to. After I had created the instruction set function and then created the read and write functions that would be used primarily to talk to the memory to receive or provide information. Finally one of the last things that I had created for this class was the interrupts and the timer. The program was the one that would usually call for the interrupt so whenever that instruction it would change the state of the CPU and memory which was also piped to the memory. The interrupt that was done by timer was given to the program by the user and as such I got it as an argument to the CPU program execution. After this I would increment a counter variable whenever the CPU piped in a new instruction and executed it. When this counter tied that of the timer it would then enter into the kernel mode and restart the counter. For the most part that was what was mainly done to implement what was need for the CPU class.

When it comes to personal experience I felt that this project, which was to create a CPU and Memory simulation allowed me to get a better more firm understand of how the CPU and memory work individually and together as well as providing me with an opportunity to program a different type of assignment. For me I had never really created a process within a Java program before and as such being able to encounter it and program with it within this project I believe allowed me to get a better understand of both the concepts behind Java processes as well as how to create and execute them. When it comes to the architect aspect as well I also believe I learned a lot in that area as well. Before doing this project I had only really know what what the CPU and memory does within a conceptual area, just learning about them through read a chapter and listening to a lecture on the topic. But from my experience doing this project and building this simulation and was able to learn in a more hands on manner much more of the intricacies of how the CPU executes instructions and how it subsequently talks to the Main Memory or RAM and vice versa as in how the main memory functions and of it also talks with the CPU.