The purpose of the project was to develop a simulated machine or computer through a CPU and Memory written with code. I believe this project has a very meaningful value to us as programmers since it teaches us how the hardware of our computers interact with each other to allow us to use it. It is always important for people to understand the tools they use and how they work so we can be more effective in reaching our goals. As a programmer, not knowing how the internals work would be similar to an aerospace engineer who does not know how an aircraft’s engine works. They work on airplanes and jets for their occupation, and we work on computers for our work. But, I would say the most important reason for us to know how the hardware works is for the errors in our code. “java.lang.OutOfMemoryError” is a fairly common error java programmers have to deal with. It happens when there is insufficient heap space. This correlates to the project since we have a set amount of memory that we can use in correlation with the CPU. Inputting more than the given amount would cause an error, just like in our java compilers.

The project was implemented with interrupts, processes, and a memory and CPU simulation. The memory has 2000 integer entries, and supports read and write operations. It reads a specified input file with a program into its array. It then uses the read operation which returns the value in the given address, and the write operation which writes data to the address. Whenever there is a ‘.’ read before a number, then the system jumps to the address. The rest of the numbers are considered for input. So overall, the memory takes care of simple commands and keeps/maintains the data given from the CPU. It talks back and forth with the CPU to make this sort of virtual machine work.

The purpose of the CPU class is to keep note of the different modes like user and system, execute various operations, and the logical and I/O (input/output) commands. It consists of several registers that correlate to the ones in an actual CPU. Some registers included are the accumulator, which is a general processing register, the instruction register, which holds the current instruction being executed/decoded, and the program counter, which keeps track of the address of the current instruction being executed. Now, for the most important part of the project, the processes. With runtime.exec, I was able to make a process that calls the memory class with arguments to determine where the file is located. The CPU also has read and write methods to send read requests, and write data to a given address.

Overall, I would say this project was quite a challenge for me. During UNIX, I struggled quite a bit with forks and pipes which translated to Java with processes. I understand the concepts but the implementation portion requires a lot more brain power from me than it should. This project has definitely opened my eyes when it comes to programming and how the internals of computers work. Even though I am not necessarily interested in making this into a career, I did enjoy experiencing it. I would definitely not consider going into computer engineering since I’m sure they would be dealing with similar code on a daily basis. I am also glad that we were able to use Java for the project since I have been working with the language since the beginning of high school, the start of my programming education. C and C++ also make using processes so much more complicated than it needs to be, so that’s why I’ll gladly be sticking to java.