Summary:

To create multiple process and allocating memory to them and then implementing Inter process communication (IPC) between these processes so that they can communicate with each other. Pipe/stream is a way to achieve this goal. Each process work on different memory slots and are interdependent.

In this project we created a simple computer system which consists CPU(Processor) and memory, Both of them are different processes. The system bus connects the two processes and we implemented it by “stream” which can be input or output. The role of Registers like is equally important in a computer system, memory hierarchy and program execution. Low level design of operating system can be defined by explaining the role of Registers, Stacks, I/O, Interrupts, IPC or Interaction between processes (memory and CPU in our case), Instruction cycle execution etc.

I implemented this project on JAVA. In my project I created two classes each representing a process i.e Processor class and Memory class. Processor is the parent class and child class Memory. Memory process is created at Runtime. Memory of the array size 2000 is allocated at the beginning to run the processes. User Program file or input file is read and copied to appropriate locations in the memory. Processor has registers like PC – to contain address of the next instruction, IR – to contain fetched instruction, Accumulator Register – AC, X,Y and Stack pointer SP. Once memory is allocated, Instruction cycle is executed. IPC is implemented using print writer and stream reader. Processor performs predefined instructions known as OPCODE. Processor fetches instructions serially from the memory using program counter and Instruction register, then switch case performs the desired operation as per the OPCODE on the operand (which in our case is in next line of the input file). 0-999 is the memory allocated for user program and 1000-1999 is for system. Timer function is implemented for interrupt at the times which user provides so it can be called as I/O interrupts. My OS handles interrupts by storing the system state on stack(push), handles interrupt and then load back the state (values of registers).

I faced many challenges during this project like converting our program into machine level code and Interaction between the processes but it was a great learning experience where I learned low level concepts of operarting system and how actually the instruction cycle is executed or a program runs. Also I learned how using Java we can make a multiprocess operating system and implement IPC.