OS Simulation

Team Members:

Jeremy Viner

Francesco Pollizi

Hein Htet Zaw

Katie Schaffer

Simulation Specifications:

Number of jobs processed: 100

Number of major Devises

CPU: 1

Short Term Queue: 1

Long Term Queue: 1

I/O Devices : 1

I/O Queue : 1

Type of Scheduling Algorithm:

Used: First Come First Serve

On which Device: Both

Second Algorithm- The second algorithm we are going to use is a round-robin using a count of \_\_\_ . This changes the project only slightly and allowing the shorter processes to get into the system faster than the longer processes that will go to completion in the first come first serve algorithm. The round robin is only used in the processor, making the I/O device continue until the I/O burst is finished. Since we are interrupting the process in the processor, it will return to the short queue once we take the processor away. We are still using only one processor, and one I/O device.

One way to make the simulation more realistic is to create a function that randomly generates processes, which fall within the proper guidelines of the program. Then having an escape key that would stop the function from creating more processes, so that once it has finished the data would be random and not as predictable as the given data. The second way to make the simulation more realistic is to have to show a process actually being worked through, instead of having a dummy process being executed. Another way to make the simulation more realistic would be doing the computations of the processor, and I/O device in actual computational time, and not counted by a clock that clicks after every round of instructions. This would give a more accurate measure to how fast the processor, and I/O actually compute in comparison to the clock we are creating to monitor “time”.