Operating System Project 1 Report

Zachary Golla

Throughput:

The results shown are expected results.

FCFS’s throughput maxes at about lambda 16. This is when the lambda matches the average service time of each process. Because each process gets its allotted amount of time, CPU utilization will always be 100% and on average it will always take 16 process to complete per second.

SRTF has a throughput that exceeds 16 processes per second because it handles all process with shorter times first and all process with longer times at the end. This causes throughput the be very high because so many processes are completing at first.

HRRN also causes a high output; however, as we will see later it removes the starvation effects of SRTF and gives better balance between longer and shorter processes.

RR behaves with a similar throughput to FCFS because each process is allotted the same amount of time per the quantum.

Average Turnaround Time:

The outcome here is as expected with some discrepancies in RR-q=.01:

FCFS sees a very low average turnaround time as long as the value of lambda is lower than or equal to average service time. At the point that the rate each process is arriving surpasses the rate that each process is finishing, then the number of process in the ready queue has increased. The higher the average number in the ready queue, then the longer each process will have to wait until is can get service; therefore, we see an increasing average turnaround time after lambda reaches 16 (1/16 is the average service time for each process).

SRTF and HRRN we see a similar result. In SRTF we choose processes with the shortest time first; therefore, as the process come in, if their times are short, they are almost immediately serviced. For HRRN, we choose the process that is shortest and has been waiting the longest, causing an extremely low average turnaround time.

Round Robin treats each process equally. Whether a long process or a short process arrives, it will be given the same amount of time to complete. So, it will have a similar average turnaround time to FCFS.

I have attempted to debug my discrepancies for RR with a low quantum. The result should be almost exactly that of FCFS and RR with a high quantum, but it is slightly lower. This is mostly due to an incorrect clock time possibly being reported.

CPU Utilization:

CPU Utilization has an expected result with a discrepancy in RR-q=.01:

At the moment that our average arrival rate surpasses our average service time the CPU will always be utilized. This is because at this moment there are always process in the ready queue. Thus, as long as the CPU has a process that need to be completed, it will be in full 100% use.

I am assuming there is a discrepancy in my clock for my RR-q=.01, but it should match all other schedulers.

Average # of Processes in the Ready Queue:

The results for average # of processes in the ready queue are as expected:

FCFS and RR will have an extremely high average number of process in the ready queue. For FCFS, each process, once it begins processing in the CPU, gets its fully allotted service time to complete and exit before the next one starts. This causes the number in the ready queue to increase dramatically once the average arrival rate of processes surpasses their average time to complete. RR gives each process the quantum amount of time to work through its process, causing all processes to switch repeatedly in and out of the CPU and ready queue. This gives the similar number of processes in ready queue to RR as FCFS.

HRRN completes shorter processes that have been waiting longer first. This means the shorter processes quickly go in and out of the ready queue while longer ones stay. This result in a very low average number of processes in the ready queue.

SRTF take the processes that are shortest and finishes those first. These results are also an extremely low average number of processes in the ready queue and are similar to HRRN.