Jesse Hunt
Chris Sigmund
CSCI 340
Dr. Leclerc

Scheduling Algorithms

In this paper, we implemented a process scheduling simulator for non-preemptive scheduling algorithms. A non-preemptive scheduling algorithm is an algorithm that allows, in this case a process to run until completion prior to the next process being allowed to run. There are three algorithms that we used in this project, First-Come-First-Served (FCFS), Shortest Process Next (SPN), and Highest Response Ratio Next (HRRN).

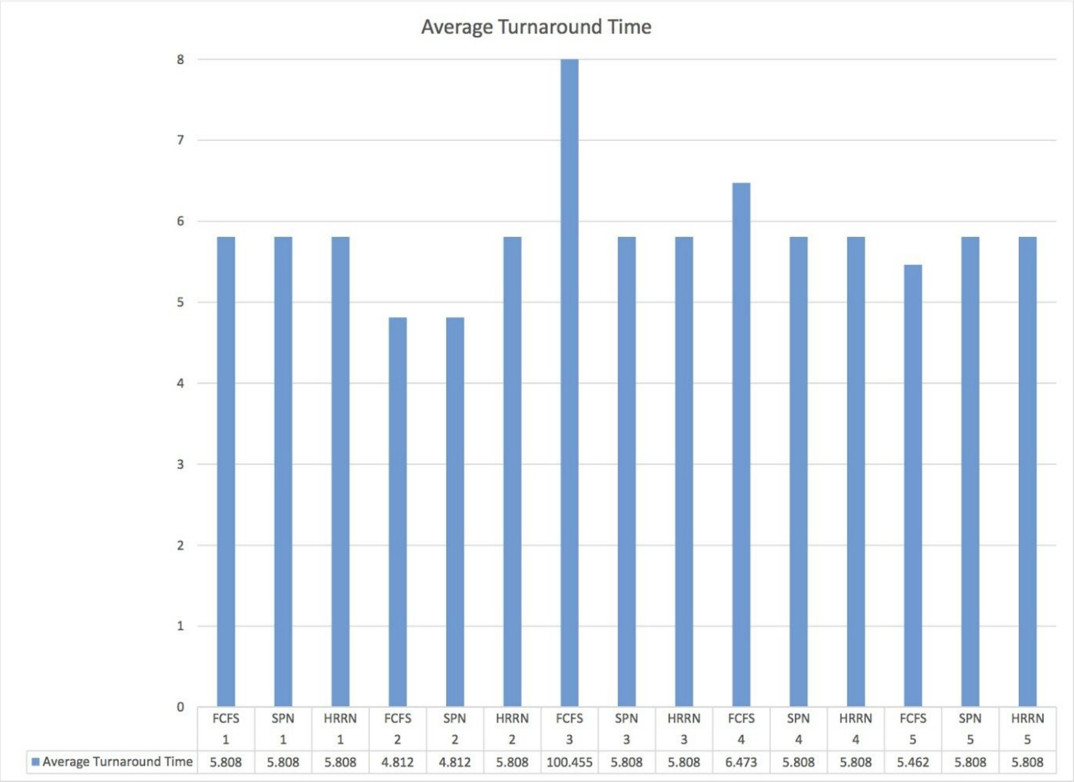
The first algorithm to test is the First-Come-First-Served (FCFS). This is considered the simplest scheduling policy, because it runs just as the name suggests. This first process that gets into the queue first will run first, and so on. This algorithm is not the most efficient for time purposes and may take longer than the other two algorithms, but all of the processes will run in order of arrival. The second algorithm is the Shortest Process Next (SPN). This algorithm will always take the process that has the shortest wait time next. This algorithm will execute processes faster than the other two algorithms, but if processes continue to be added to the queue, then there is a chance that longer processes will never run, thus creating starvation. The third algorithm is the Highest Response Ratio Next (HRRN). This algorithm calculates the response ratio, which is time spent waiting for the processor plus the expected service time, divided by the expected service time. This allows for the process waiting for longer periods of time to take precedence over shorter processes that will finish faster. This algorithm will not allow for starvation, unlike the SPN, but can change the order of processes, by recomputing the response ratio after each process completes execution.

In this simulation, the two performance metrics used to determine the overall performance of the three algorithms is the turnaround time and the wait time. The turnaround time is the interval of time between the submission of a process and its completion. It includes actual execution time plus time spent waiting for resources, including the processor.

Turnaround time is calculated by subtracting the time of completion of an executing a process and the arrival time of the process. Another performance metric is the wait time. The wait time is computed by current time of execution minus the arrival time. The average wait time is a decent performance metric because depending on the algorithm being used, starvation can occur and the larger the average wait time for an algorithm it can then be considered very inefficient.

To create our simulation, we use a heap and array. First, we created an array of random processes, and for each of the three algorithms we initialized a heap to sort the processes based on the algorithm's requirements. We then loop through the processes in the array, as long as the process has not been taken or there are processes in the array or the heap is not empty, the loop will continue. During this loop, another loop is created that goes through all the number of processes created in the array. In this loop, if a process is not taken and the total time of the execution of the algorithm is greater than the arrival time of the process the process is added to the heap and removed from the array. Once this loop ends, if the heap is empty one is added to the total time of the algorithm and one is added to the age of the heap. Otherwise, the minimum process is removed from the heap and the total time is added to itself plus the service time of the process. Then the loop ends after the while statement fails and results are printed to a csv file. The process is then repeated for the other two algorithms and the results are sent to the same csv file.

While reviewing the results of our simulation, we found that there are a major discrepancy with one of the seeds during one of our First Come First Served simulations. We found that the Poisson distribution included many processes with zero service time as well as added many processes at the same time, so it ballooned our wait times and caused an abnormality in our results. Consistently, most efficient average turnaround time was Highest Response Ratio Next, while there were instances when the FCFS and SPN performed better, however those results were not typical.



	FCFS 1	SPN 1	HRRN 1	FCFS 2	SPN 2	HRRN 2	FCFS 3	SPN 3	HRRN 3	FCFS 4	SPN 4	HRRN 4	FCFS 5	SPN 5	HRRN 5
■ Average Turnaround Time	5.808	5.808	5.808	4.812	4.812	5.808	100.455	5.808	5.808	6.473	5.808	5.808	5.462	5.808	5.808