Matt Adamson, James Shockley
4/18/16

Scheduling Algorithm Simulation

Scheduling Algorithms:

For this project, three process scheduling algorithms will be compared in efficiency. The First Come First Serve (FCFS), Shortest Process Next (SPN), and Highest Response Ratio Next (HRRN), each have benefits and drawbacks in terms of process execution.

Performance Metrics:

Two performance metrics used in the experiment are turnaround time and waiting time. Turnaround time is the time from when a process arrives to the when the process is finished. Waiting time is the amount of time a process waits in the system before it starts to be processed.

First Come First Serve:

The FCFS operates as a queue of processes, executing each process in order of arrival time. This method is similar to a shopping mall, in which customers check out in a line regardless of how long each item takes to pass.

A main advantage of FCFS over other algorithms is that processes in FCFS never encounter starvation since smaller processes can never cut in front of

larger processes. FCFS also has a smaller overhead compared to SPN and HRRN.

However, FCFS is unfriendly towards processes with short service times. In the Table 1 below, the third process has a service time of only 1 unit, but has a high turnaround time of 151 units. This high waiting and turnaround time is caused by large processes holding smaller processes back. In a real FCFS system, these larger processes often restrict I/O processes, lowering the efficiency of the operating system.

First Come First Serve Scheduling Example				
Arrival Time	Service Time	Waiting Time	Turnaround Time	Finish Time
0	100	0	100	100
50	50	50	100	150
100	1	50	51	151
150	200	1	201	351
200	150	151	301	501
250	20	251	271	521
Average Waiting Time: 83.83				
Average Turnaround Time: 170.66				

Table 1: First Come First Serve Scheduling

Shortest Process Next:

The SPN algorithm always executes the process with the shortest service time. However, due to arrival time differences, the shortest job does not always go first.

SPN's advantage over FCFS is that SPN favors short processes, drastically reducing wait time for queues containing many quick jobs. In Table 2 below, several of the processes have almost no waiting time due to their small size and are executed almost instantly.

However, SPN's weakness is that large processes can starve if there exists an indefinite amount of small processes.

Highest Response Ratio Next:

Similar to SPN, he HRRN scheduling algorithm gives priority to processes with shorter run times. However, HRRN eventually gives large processes priority over smaller ones after a period of time to prevent starvation. However, HRRN's disadvantage to SPN is that since priority can be given to large processes, the waiting and turnaround times are slightly higher than SPN.

As HRRN is a priority scheduler, processing runs faster than the simple FCFS. A disadvantage of HRRN though is that this algorithm is has a higher overhead than FCFS.

Shortest Process Next Scheduling Example				
Arrival Time	Service Time	Waiting Time	Turnaround Time	Finish Time
0	100	0	100	100
100	1	0	1	101
50	50	51	101	151
150	200	1	201	351
250	20	101	121	371
200	150	171	321	521
Average Waiting Time: 54.00				
Average Turnaround Time: 140.83				

Table 2: Shortest Process Next Scheduling

Highest Response Ration Next Example				
Arrival Time	Service Time	Waiting Time	Turnaround Time	Finish Time
0	100	0	100	100
50	50	50	100	150
100	1	50	51	151
150	200	1	201	351
200	150	101	121	371
250	20	171	321	521
Average Waiting Time: 62.17				
Average Turnaround Time: 149.00				

Table 3: Highest Response Ratio Scheduling

FCFS Simulation Implementation:

The FCFS algorithm used a simple for loop to iterate through the inputted arrays of arrival and service times. Each index's waiting time was set to the previous index's finish time minus the arrival time.

The current turnaround and finish times were also calculated inside of the for loop.

Next, for simplicity, a second for loop calculated the sums of turnaround and waiting time. The function then returned the average waiting and turnaround time.

SPN Simulation Implementation:

The SPN first finds all of the available processes that can be executed. Next, a selection sort function is called to sort these available processes by smallest to largest (the shortest jobs go first). Once the processes are in order, a for loop iterates through that array to calculate the waiting and turnaround times.

HRRN Simulation Implementation:

The HRRN algorithm is another non-preemptive algorithm similar in implementation to SPN, but without the greater potential for starvation among larger service time processes through a tradeoff in service and wait times since those jobs compete against those that have shorter run times. The implementation itself consists of assigning a priority after each job exceeds its time-slice using the formula:

$$\text{priority} = \frac{waitingTime + burstTime}{burstTime}$$

This equation results in a response ratio unique to the process at that time. Jobs gain higher priority the longer they are waiting, as time relative to arrival $t \to \infty$ the response ratio implies inevitability. Originally, we sought out to use data structures of our own design, a linked list consisting of process nodes but sought a simpler implementation utilizing sequences of arrays controlled by a common index representing the process id.

Simulation Times:

For all three algorithms, simulation time was calculated in nanoseconds, clocking in right before the method was called and clocking out after the algorithm finished.

Algorithm Results:

Based on Figures 1 and 2, SPN has the shortest wait and turnaround times. FCFS has such a high time compared to SPN and because the FCFS lags on small processes. HRRN has a time between FCFS and SPN because it speeds through short processes but also gives priority to some large processes, slowing down the turnaround and wait times.
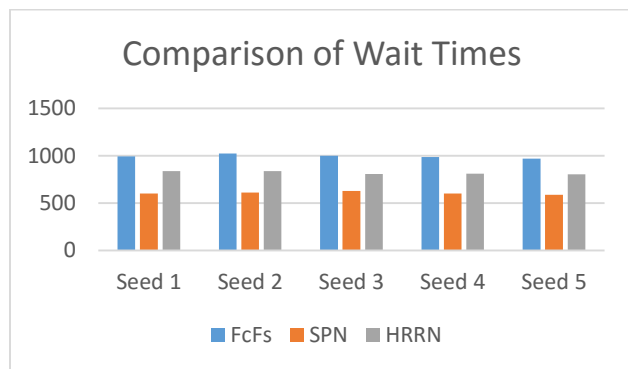
Comparison of Turnaround Times

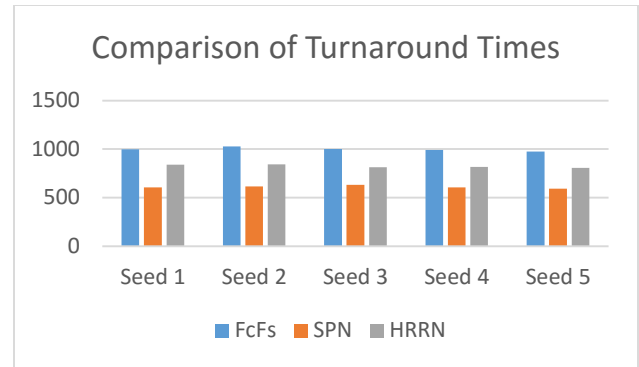Figure 2: Turnaround times for arrays of 1000 processes.

Comparison of Wait Times

Figure 1: Wait Times for arrays of 1000 processes.

The differences between the turnaround and wait times are very small (only about 5 units) because each of the simulated process times had a value between 0 and 10. On the previous tables, the service times ranged from 1 to 200, to give a larger difference between wait and turnaround times.