Santa Clara University

COEN 383 - Advanced Operating System

Professor: Ahmed Ezzat

Winter 2021

Group 5: Anh Truong, Quan Bach, Travis Le, Yukun Zhang, Pauldin Bebla

PROJECT 2 REPORT

First Come First Serve Algorithm Analysis

First Come First Serve is the easiest algorithm to understand and implement. There is only one queue, and the job at the head of the queue will be processed until it's done. This causes starvation problems if the jobs at the beginning of the queue have very long service time. The throughput tends to be on the bad side, and similar to HPF-NP, long jobs at the beginning of the queue decrease the number of scheduled jobs. Average response time is also very high because jobs have to wait for a very long time before they get the CPU. For the same above reason, turnaround time and wait time are also very high. As a result, FCFS is not a fair nor efficient algorithm .

Round Robin Algorithm Analysis:

In the implementation of Round Robin scheduler, the time slice for preemptive condition is 1 quanta. This has a big impact on the response time. It's highly likely that jobs will be scheduled since no job will have the CPU for more than 1 quanta. Similarly, wait-time is not too high as expected. However, this affects the Turnaround time greatly since jobs with large burst time will keep getting preempted and pushed back in the process queue. With only one queue and small preemptive time slice condition, starvation will almost never occur. Thus, this scheduler is good at fairness but not as great at efficiency in a short window of time (i.e. analysis in 100 qantas).

Highest Priority First - Non-preemptive Analysis:

In Highest Priority First Non-preemptive, the average turnaround time and average wait time will have the same value. This is the case when comparing a single queue's turnaround time and same queue's wait time.This is because when a job is being processed, the CPU will not switch to another job when it's running. In theory, the values of the first queue should be roughly similar to FCFS except somewhat worse. However, the data indicates that it's slightly better than FCFS.

This algorithm has to take in 4 queues rather than just one singular queue. The 1st queue has lower values than all the other queues because it is the first one being processed. So, the other queues will have to wait longer for their jobs to be processed. In some cases, the 2nd, 3rd, 4th queue ends up being starved because there are too many jobs assigned to the 1st priority. In some cases concerning the output, the 2nd could be starved but the 3rd one is not.

The 1st queue will not be starved of jobs unless there are no jobs belonging to the first priority queue. Processes in the highest priority queue (1st) have the best response and wait time. The turnaround time seems pretty decent compared to other schedulers. The 4th priority queue has the worst average time, worst turnaround time, and worst wait time because processes are unable to get their jobs done. In total, it has a not great response time and wait time. Turnaround time is not good also in comparison to SJF, SRT, and RR.

Since the queues are split up, the throughput of each individual queue is worse than all the other algorithms' throughputs. Total throughput of the whole queue is very similar to FCFS. In other words, it has a bad throughput.

Highest Priority First - preemptive analysis:

In the implementation of Highest Priority First with preemptive by Round Robin, the performance of queue-1 i.e. the top priority queue is a lot better than other queues. However, this has caused starvations in the lower priority queues such as queue-3 and queue-4. Starvation happened consistently in queue-4 across many test runs. This implementation has proven that it's optimal to use HPF-preemptive when scheduling interactive jobs with higher priority than others.

Shortest Job First Analysis:

Shortest Job First algorithm is a non preemptive algorithm that uses a queue for arrived jobs and sorts them by service time. Whenever a job is finished in the CPU, the job with the shortest service time in the queue will be the one entering the CPU. Because of non preemptive scheduling, the average response time and average wait time are equal. When choosing the appropriate job for the CPU, the Shortest Job First algorithm always chooses the job with the shortest service time and this job will be finished in a short time so other jobs will not wait for too long. To sum up, this is the best approach to minimize wait time.

Shortest Remaining Time First Analysis:

Shortest Remaining Time algorithm is preemptive, and is a variation of SJF. At the beginning of each time slice, the algorithm checks if newly arrived processes have a shorter service time than the process at the front of the queue. If so, it re-sorts the current process queue by least remaining time, and continues to run the process at the front of the queue for one quantum. If a process gets the CPU, the only way it can give it up is if either the process finishes or a newly arrived process has a shorter remaining time, so the average wait time is low. Although the average turnaround time is low overall, processes with long service times are subject to starvation if a lot of shorter processes arrive afterward. Similarly, long processes can skew the overall average response times upward if they enter a queue already full of shorter processes. Overall, it's an efficient algorithm if you know all the service times beforehand.

Comparison Between Algorithm Performance

SJF and SRT perform really well overall. Their algorithms are very fair and efficient. FCFS is an easy-to-implement algorithm but comes at the expense of very bad overall metrics. Round Robin has a very good balance between fairness and efficiency. Its throughput is the best of all algorithms. HPF-NP has similar metrics for response time, turnaround time, and wait time as the Round Robin but the throughput is not as high. HPF-P has the best turnaround time and wait time. Its throughput and response time comes in the middle of all the other algorithms.

Green colors indicate the best of all.

Red colors indicate the worst of all.

	FCFS	**SJF**	**SRT**	**RR**	**HPF-NP**	**HPF-P**
Average Response Time	29.45	2.79	7.21	13.12	12.84	12.58
Average Turnaround Time	35.38	7.25	10.39	15.16	18.26	2.51
Average Wait Time	29.45	2.79	6.59	14.68	12.84	2.41
Average Throughput	0.17	0.23	0.28	0.36	0.19	0.17