Computer Engineering 571: Embedded Operating Systems

Programming Assignment 2

Skye Russ

The sample algorithm implements a basic Round Robin scheduling algorithm. To implement First Come First Serve, the loop was removed and each process was allowed to run until “wait_pid” returned that the process completed. To implement Shortest Job First, the processes were sorted by workload before being executed in the same way as First Come First Serve. To implement Multiple Level Feedback Queue, the initial Round Robin implementation was used with the loop removed, then the First Come First Serve implementation was copied after each process executed for one time quantum.

Each implementation assumed that the processes would not be killed by any other process in the system during execution. Furthermore, all processes were deemed to arrive to the scheduler at the start of execution in ascending numerical order at the same time.

To allow the schedulers to work efficiently, a “process description” struct was created. This struct stored the information detailed in Table 1. An array of this struct was used to sort the algorithms for Shortest Job First and used to keep track of information about each process as it executed in each scheduling algorithm.

Table 1: Process Description Struct Definition

Data Field	**Type**	**Notes**
PID	pid_t	The PID of the process
WORKLOAD	int	The workload of the process
ID	int	The order the process was created (for FCFS)
PROCESSING TIME	double	The current processing time the process has taken.
END TIME	struct timespec	The time the process finished its execution.

Because each process entered the system at the same time, timing was started after each process had been created and frozen. The Response time was calculated by taking the processes’ end time and subtracting it from the start time to get the difference in seconds.

The context switch overhead is calculated by subtracting the total processing time (stored in the Process Description for each task) from the end time of the last process to complete. A timing marker was taken after each scheduling algorithm detected that every process was completed which marked the end time of the script.

For the following experiments, each script was executed 100 times. After each script execution, the average response time and context switch overhead were recorded in a CSV file. The means and standard deviations of all 100 iterations were calculated using MATLAB and recorded in a table using the “create_graphs.m” script. Differing standard deviations were recorded for each case with Round Robin having the highest by a large margin, followed by Shortest Job First, then Multiple Level Feedback Queues and First Come First Serve. Furthermore, Round Robin had the highest context switch overhead by a large margin. These two factors likely indicate that there is more variance when more context switches happen for the scheduler.

Experiment 1: Best time quantum value for Round Robin

The best time quantum value recorded for Round Robin was 17,000ms. Initially, time quantum values were tested in increments of 5000ms from 0 to 100,000ms. After calculating the average response time of all iterations among all time quantums, the values that were less than 20,000ms had the best average response time. The workload was executed again with new time quantum values tested in increments of 500ms from 0 to 20,000ms. Through this higher granularity, the best time quantum value was found to be 17,000ms closely followed by 500ms.

The average response time changes based on different time quantum value because certain processes will finish their execution faster and therefore decrease the total average response time. Because of this, it would likely not significantly improve average response time by having specific time quantums for specific processes unless those quantums exactly matched the execution time of the shorter processes. If the values exactly matched the execution time of the shortest processes, the algorithm would essentially be shortest job first which has the lowest average response time of all algorithms (seen in Experiment 3).

As expected, when lower time quantum values were used, the context switch overhead increased proportionally. The difference in average response time between 500ms and 17,000ms was in the order of thousandths of seconds, however after 100 iterations 17,000ms had a lower mean average response time.

Experiment 2: Best time quantum value for MLFQ

The best time quantum value recorded for the Multiple Level Feedback Queue scheduling algorithm was 84,000ms. Initially, time quantum values were tested in increments of 5000ms from 0 to 100,000ms. After calculating the average response time of all iterations, two local minimums were detected. Because of this, the script was run again with increments of 1000ms from 15,000ms to 32,000ms and 81,000ms to 100,000ms.

Furthermore, the usage percentage of the second queue was recorded during these tests (seen in Table 2). It is notable that the algorithm was nearly identical to First Come First Serve when quantum values were less than 15,000ms and always had at least two processes passed to the second queue after the first round of round robin.

Table 2: Second Queue Usage Percentage

Quantum Value	Second Queue Usage Percent
15000	0.9775
17000	0.75
84000	0.5

It would likely help to have different time quantum values for each process if and only if those time quantum values exactly matched the execution time of the shortest processes. In that case, the shortest processes would never be passed to the second queue and be forced to wait for the longest processes to complete their execution, thus greatly reducing the average response time. If the quantum value for the short processes matches the execution time, the multiple level feedback queue algorithm would very closely match a shortest job first algorithm in the first queue rather than a round robin algorithm.

The context switch overhead for this algorithm remained low, but notably still higher than First Come First Serve and Shortest Job First. Notably, the context switch overhead slightly correlated with the number of processes that were passed to the second queue. This is because the amount of context switches required directly correlates with the number of processes that must be executed in each queue.

Experiment 3: Comparing the performance of all algorithms:

When the ideal time quantums are used, the scheduling algorithms have the average response time recorded in Table 3. Notably, the Shortest Job first algorithm had the lowest average response time of all the algorithms. It also managed to have a comparable context switch overhead to First Come First Serve. This is because there were a total of three context switches in each of those algorithms.

This supports the theoretical outcomes that we studied in class. We know that Shortest Job First is faster than First Come First Serve and Round Robin and the Multi-Level Feedback Queue algorithm is simply a combined First Come First Serve and Round Robin implementation.

Table 3: Scheduling Algorithms Compared (Different Workloads)

Scheduling Algorithm	Average Response Time	Standard Deviation	Context Switch Overhead
Multi-level Feedback Queues (QUANTUM = 84000)	0.9247	3.4349e-04	6.0238e-05
Round Robin (QUANTUM = 17000)	0.6703	0.0099	3.7209e-04
First Come First Serve	1.4226	4.8880e-04	7.7667e-06
Shortest Job First	0.5167	3.0762e-04	7.6857e-06

When the workloads were changed to be identical (100,000), the following average response times were recorded (Table 4). Each case was repeated 100 times and then the average response times were averaged to get the values in the table. Notably, there is very little difference between all the cases except for Round Robin. Round Robin has a significantly higher context switch overhead as well as a much higher average response time.

Table 4: Scheduling Algorithms Compared (Same Workload)

Scheduling Algorithm	Average Response Time	Standard Deviation	Context Switch Overhead
Multi-level Feedback Queues (QUANTUM = 84000)	2.9825	0.0010	2.1020e-05
Round Robin (QUANTUM = 17000)	4.6473	0.0126	6.2957e-04
First Come First Serve	2.8568	0.0018	8.4303e-06
Shortest Job First	2.8559	0.0007	1.0428e-05

Compared to the other case, all algorithms perform significantly worse. Furthermore, Round Robin falls to the worst algorithm, and the other algorithms are slightly different from each other. Notably, Shortest Job First is almost identical to First Come First Serve because the sorting orders the algorithms are in the same order as they came in. Furthermore, the standard deviation of Round Robin is an order of magnitude above the other cases. This is because there are significantly more context switches during its operation.

It would be prudent to recalculate the ideal time quantum values compared to 3a. The Multiple Level Feedback Queue algorithm always sends all four processes to the second queue which makes it behave almost the same as First Come First Serve with double the context switches.

As recorded before, the context switch overhead is significantly higher for the Round Robin algorithm compared to the other algorithms, however, the time is still a minuscule fraction of the response time and execution time.