Program 2 Report

By: Misha Ward

Program 2 Report

By: Misha Ward

How to Run Programs:

In order to run, please save the zip file and locate files in terminal. You may need to rename the files to work with Thread OS. Rename the Scheduler_part1, or Scheduler_part2 to Schedule and place into the folder with ThreadOS. At this point, locate the Thread OS file with terminal and type the command: **javac Scheduler.java** and then run the file using: **java Scheduler** .

In order to run with ThreadOS, type into terminal: **java Boot** then: **l Test2** or: **l Test2b** .

Design and Algorithm:

The multilevel feedback queue scheduler works as the following:

First, the algorithm creates three queues that holds the various threads and also initialize integers that represents the count of time that the thread will take. Next, the scheduler run and select the first queue to execute threads. The scheduler will check queue 0 for a threads. While there are more than one thread, the algorithm will go through the threads and cause that thread to resume. Once the thread runs, the code asks for the thread to sleep for 500 ms. It then goes and synchronizes the queue and suspends the operation if the thread is not null and is active, suspend the thread, otherwise remove the thread and move it to the second queue.

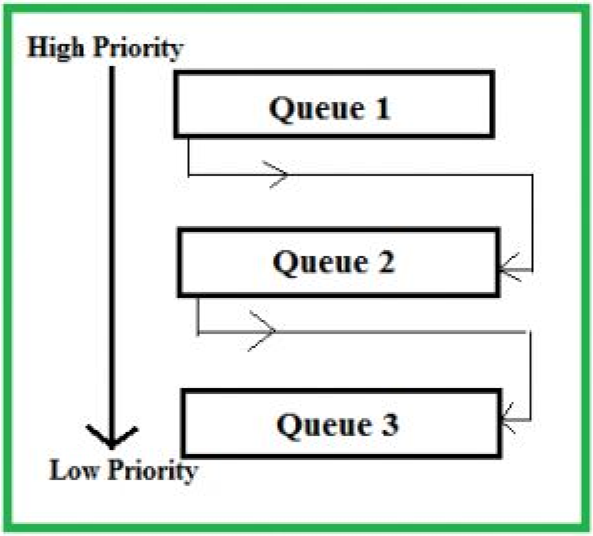
The process is very similar for queues 1 and 2 with slight differences:
For queue 1, the algorithm has the thread sleep for 500 ms, then checks the first queue for any other threads. If the first queue has threads, the algorithm suspends the current thread and repeats the first queue execution. Once those have been completed, it resumes the thread, then sleeps the thread for another 500 ms, then synchronizes the queue. If the current thread at that point is not null and also alive, the algorithm suspends, otherwise the thread is removed from the queue 1 and moved into the queue 2.

Finally, for the last queue (queue 2), the algorithm is the same except for the process wait. At this point, the algorithm has the thread sleep while checking if the first or second queues have threads, if they do, the current thread is suspended and it executes the first queue and then the second queue. Finally, the process resumes and this repeats 4 times. Finally the third queue is synchronized and if the current thread is not null and is active, the thread is suspended, otherwise the the thread removes the current thread and places it back into the queue.

Below is a simple graphic showing the process flow of data for the queues.

Program 2 Report

By: Misha Ward



Results Comparison:

Overall, I believe that the MFQS algorithm is more efficient as the algorithm manages the system resources better. In the red robin version, the algorithm does many context switches which were unnecessary. MFQS runs a lot of the threads in quick bursts at the start but eventually lets threads run longer as the scheduler runs. One can see that all the averages are much better for MFQS than the Red Robin when running Test2. Turn around time is better for MFQS due to the fact that shorter processes finish quickly while longer ones will eventually get to finish. By allowing shorter processes to finish, it limits their turnaround time as well as the longer ones. Finally, response time is also better with MFQS as the shorter bursts at the start helps speed up the response for the shorter processes which also shortens the response times for the longer processes. Overall, MFQS should be used instead of Red Robin scheduling.

Red Robin			
Thread	Response Time	Turnaround Time	Execution Time
a	1998	29091	27093
b	3002	10020	7018
c	4006	21055	17049
d	5008	33105	28097
e	6010	6528	518
Average	**4004.8**	**19959.8**	**15955**

Program 2 Report

By: Misha Ward

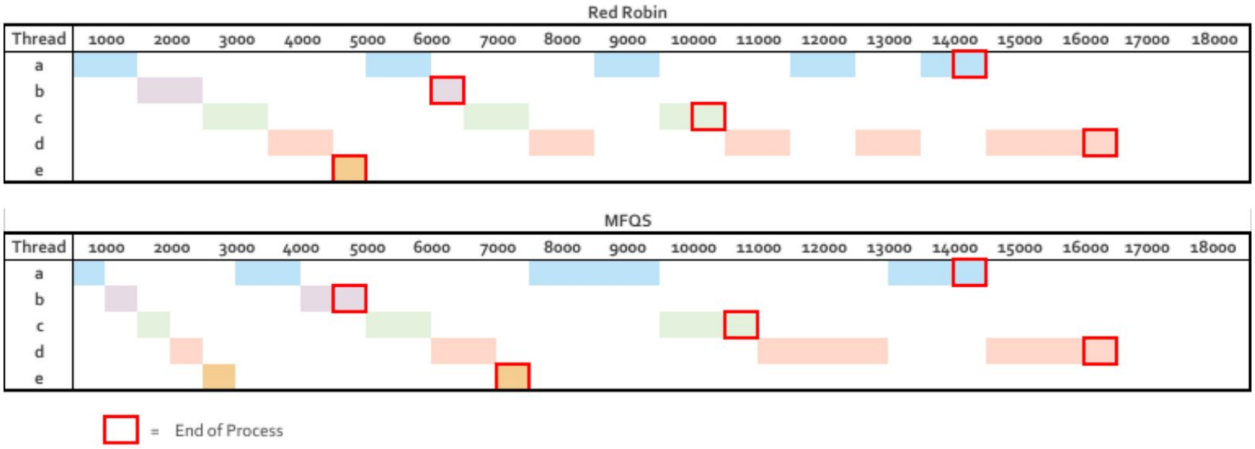
MQFS			
Thread	Response Time	Turnaround Time	Execution Time
a	498	23198	22700
b	1002	5552	4550
c	1507	15140	13633
d	2009	30213	28204
e	2513	8048	5535
Average	**1505.8**	**16430.2**	**14924.4**

First Come First Serve Hypothesis:

I feel that this depends entirely on what the processes that are coming into scheduler and looking at these burst speeds for the threads, the FCFS will cause the response time to increase while the execution time will decrease. The reason the execution time will decrease is that there are no interrupts and all processes will just be completed one after another. The issue with this is if there are huge processes, they will hog the computer for a long time even if there are simple or other critical activities that need to be completed.

Gantt Chart:

Red Robin and MFQS Gantt Chart:



Preliminary Results:

Part 1: Red Robin

l Test2

threadOS: a new thread (thread=Thread[Thread-5,5,main] tid=1 pid=0)

threadOS: a new thread (thread=Thread[Thread-7,5,main] tid=2 pid=1)

```
threadOS: a new thread (thread=Thread[Thread-9,5,main] tid=3 pid=1)
threadOS: a new thread (thread=Thread[Thread-11,5,main] tid=4 pid=1)
threadOS: a new thread (thread=Thread[Thread-13,5,main] tid=5 pid=1)
threadOS: a new thread (thread=Thread[Thread-15,5,main] tid=6 pid=1)
Thread[e]: response time = 6010 turnaround time = 6528 execution time = 518
Thread[b]: response time = 3002 turnaround time = 10020 execution time = 7018
Thread[c]: response time = 4006 turnaround time = 21055 execution time = 17049
Thread[a]: response time = 1998 turnaround time = 29091 execution time = 27093
Thread[d]: response time = 5008 turnaround time = 33105 execution time = 28097
```

Part 2: MFQS

```
l Test2
threadOS: a new thread (thread=Thread[Thread-5,5,main] tid=1 pid=0)
threadOS: a new thread (thread=Thread[Thread-7,5,main] tid=2 pid=1)
threadOS: a new thread (thread=Thread[Thread-9,5,main] tid=3 pid=1)
threadOS: a new thread (thread=Thread[Thread-11,5,main] tid=4 pid=1)
threadOS: a new thread (thread=Thread[Thread-13,5,main] tid=5 pid=1)
threadOS: a new thread (thread=Thread[Thread-15,5,main] tid=6 pid=1)
Thread[b]: response time = 1002 turnaround time = 5552 execution time = 4550
Thread[e]: response time = 2513 turnaround time = 8048 execution time = 5535
Thread[c]: response time = 1507 turnaround time = 15140 execution time = 13633
Thread[a]: response time = 498 turnaround time = 23198 execution time = 22700
Thread[d]: response time = 2009 turnaround time = 30213 execution time = 28204
```

Appendix:

Red Robin: Test2b

```
threadOS: a new thread (thread=Thread[Thread-17,5,main] tid=7 pid=0)
threadOS: a new thread (thread=Thread[Thread-19,5,main] tid=8 pid=7)
threadOS: a new thread (thread=Thread[Thread-21,5,main] tid=9 pid=7)
threadOS: a new thread (thread=Thread[Thread-23,5,main] tid=10 pid=7)
threadOS: a new thread (thread=Thread[Thread-25,5,main] tid=11 pid=7)
threadOS: a new thread (thread=Thread[Thread-27,5,main] tid=12 pid=7)
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[b] is running
```

Program 2 Report

By: Misha Ward

```
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[e] is running
Thread[e] is running
Thread[e] is running
Thread[e] is running
Thread[e] is running
Thread[e]: response time = 6016 turnaround time = 6534 execution time = 518
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
```

Program 2 Report

By: Misha Ward

```
Thread[a] is running
Thread[a] is running
Thread[b]: response time = 3003 turnaround time = 10027 execution time = 7024
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
```

Program 2 Report

By: Misha Ward

```
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[c]: response time = 4008 turnaround time = 21061 execution time = 17053
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[d] is running
Thread[d] is running
```

```
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[a]: response time = 2003 turnaround time = 29086 execution time = 27083
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d]: response time = 5010 turnaround time = 33094 execution time = 28084
Test2b finished
```

MFQS: Test2b

```
l Test2b
threadOS: a new thread (thread=Thread[Thread-17,5,main] tid=7 pid=0)
threadOS: a new thread (thread=Thread[Thread-19,5,main] tid=8 pid=7)
threadOS: a new thread (thread=Thread[Thread-21,5,main] tid=9 pid=7)
threadOS: a new thread (thread=Thread[Thread-23,5,main] tid=10 pid=7)
threadOS: a new thread (thread=Thread[Thread-25,5,main] tid=11 pid=7)
threadOS: a new thread (thread=Thread[Thread-27,5,main] tid=12 pid=7)
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
```

Program 2 Report

By: Misha Ward

```
Thread[c] is running
Thread[c] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[e] is running
Thread[e] is running
Thread[e] is running
Thread[e] is running
Thread[e] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[b] is running
Thread[b]: response time = 999 turnaround time = 5550 execution time = 4551
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
```

Program 2 Report

By: Misha Ward

```
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[e]: response time = 2514 turnaround time = 8050 execution time = 5536
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c] is running
Thread[c]: response time = 1504 turnaround time = 15152 execution time = 13648
Thread[d] is running
```

```
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a] is running
Thread[a]: response time = 498 turnaround time = 23199 execution time = 22701
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
```

```
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d] is running
Thread[d]: response time = 2010 turnaround time = 30208 execution time = 28198
Test2b finished
```