**COMPSCI 377: Operating Systems**

Homework 3

Due: Monday, Feb 13, 2017 at 4pm

1. Using scheduler.py, solve problem 2 from chapter 7.

SJF:  
Here is the job list, with the run time of each job:

Job 0 ( length = 100.0 )

Job 1 ( length = 200.0 )

Job 2 ( length = 300.0 )

\*\* Solutions \*\*

Execution trace:

[ time 0 ] Run job 0 for 100.00 secs ( DONE at 100.00 )

[ time 100 ] Run job 1 for 200.00 secs ( DONE at 300.00 )

[ time 300 ] Run job 2 for 300.00 secs ( DONE at 600.00 )

Final statistics:

Job 0 -- Response: 0.00 Turnaround 100.00 Wait 0.00

Job 1 -- Response: 100.00 Turnaround 300.00 Wait 100.00

Job 2 -- Response: 300.00 Turnaround 600.00 Wait 300.00

Average -- Response: 133.33 Turnaround 333.33 Wait 133.33

FIFO:  
\*\* Solutions \*\*

Execution trace:

[ time 0 ] Run job 0 for 100.00 secs ( DONE at 100.00 )

[ time 100 ] Run job 1 for 200.00 secs ( DONE at 300.00 )

[ time 300 ] Run job 2 for 300.00 secs ( DONE at 600.00 )

Final statistics:

Job 0 -- Response: 0.00 Turnaround 100.00 Wait 0.00

Job 1 -- Response: 100.00 Turnaround 300.00 Wait 100.00

Job 2 -- Response: 300.00 Turnaround 600.00 Wait 300.00

Average -- Response: 133.33 Turnaround 333.33 Wait 133.33

1. Using mlfq.py, solve problem 1 from chapter 8.

./mlfq.py -n 2 -j 2 -s 1151515151 -M 0 -m 20 -c  
  
Final statistics:

Job 0: startTime 0 - response 0 - turnaround 28

Job 1: startTime 0 - response 10 - turnaround 20

Avg 1: startTime n/a - response 5.00 - turnaround 24.00

./mlfq.py -n 2 -j 2 -s 1151515151 -M 0 -m 40 -c

Final statistics:

Job 0: startTime 0 - response 0 - turnaround 58

Job 1: startTime 0 - response 10 - turnaround 51

Avg 1: startTime n/a - response 5.00 - turnaround 54.50

1. Using lottery.py, solve problem 2 from chapter 9.

Take (random) % (number of tickets)

When the number of tickets is very imbalanced, the job with the majority of the total tickets assigned has a significantly higher chance to be chosen. With job 0 having 1 ticket and job 1 having 100 tickets, job 0 has under a 1% chance of being chosen to run at first. In general, with a huge imbalance in tickets per job, the lottery scheduler will run lower ticketed jobs with a very low probability, and has the possibility of starving the CPU from the process.

1. Solve problem 3 from chapter 9, but with job length of 50 (instead of 100)

Seed 1: job 1 done 196, job 0 done 200  
Seed 12: job 1 done 189, job 0 done 200  
Seed 5: job 1 done 181, job 0 done 200

Seed 131241: job 1 done 199, job 0 done 200.  
  
The lottery scheduler seems to tend towards being perfectly fair, where 50% of the tickets that are chosen will be for job 0 in comparison to job 1. This makes sense because there is a certain probability (1/total tickets) for one ticket to be chosen, so with 100 tickets per job, there is a 50% chance for each. job to be chosen.