PROJECT 1 REPORT
DISCREET TIME EVENT SIMULATOR

ALFONSO DE LA MORENA
A_D426@TXSTATE.EDU

TEXAS STATE UNIVERSITY
INGRAM SCHOOL OF ENGINEERING

DR. MINA GUIRGUIS
CS 4328 – OPERATING SYSTEMS
SAN MARCOS, TX

MAR. 29, 2019

1 OVERVIEW

1.1 Abstract

In this project, we are going to build a discrete-time event simulator for several CPU schedulers on a single CPU system. The goal of this project is to compare and assess the impact of different schedulers on different performance metrics, and across multiple workloads.

Contents

2 SUMMARY

The following scheduling algorithms will be implemented using python:

1. First-Come First-Served (FCFS)
2. Shortest Remaining Time First (SRTF)
3. Highest Response Ratio Next (HRRN)
4. Round Robin, with different quantum values (RR)

The following metrics will be measured for each one:

- The average turnaround times
- The total throughput (number of processes done per unit time)
- The CPU utilization
- The average number of processes in the ready queue

The tests for each f the scheduling algorithms will be performed with the following metrics:

- A lambda value varying from 1 to 30
- An average service time of 0.06
- A round robin Quantum of 0.01 and 0.2

These variables result in a total of 150 runs recorded by the simulator.

3 HOW TO RUN

3.1 Python Files

The project is written in python. It includes to main files:

1. 'sim.py' used for running individual simulation tests. It takes 4 inputs in the following order:
    1) Schedule algorithm: a number from 1 to 4 specifying which scheduling algorithm to use. (1: FCFS, 2: SRTF, 3: HRRN and 4: RR)
    2) Lambda: accepting a value from 1 to 1000
    3) Avg Service Time: accepting a value from 0 to 1000
    4) Round Robin Quantum: accepting a value from 0 to 1000

2. 'modules.py' used for defining all of the functions and objects used in the simulator.

The example commands for running one test individually is as follows:

`python sim.py` or `python sim.py 2 15 .06 .2`

3.2 Batch Files

The project also includes batch files in order to run. There is one batch file for each of the scheduling algorithms. The have the following file names:

1. 'runSimFCFS.sh': runs FCFS with lambda values from 1 to 30
2. 'runSimSRTF.sh': runs SRTF with lambda values from 1 to 30
3. 'runSimHRRN.sh': runs HRRN with lambda values from 1 to 30
4. 'runSimRoundRobin.sh': runs Round Robin with lambda values from 1 to 30 and Quantum values of 0.01 and 0.2

The example commands for running any of the batch files is as follows:

`sh runSimFCFS.sh` or `sh runSimRoundRobin.sh`

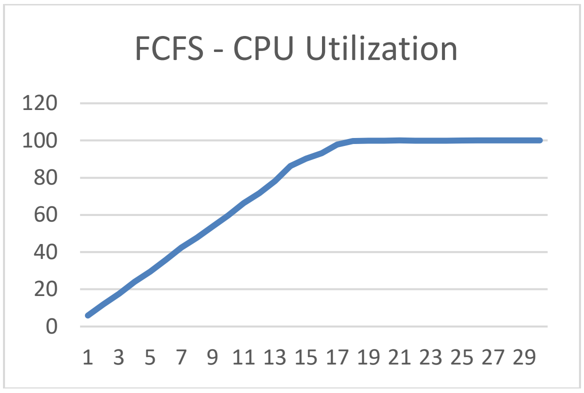
3.3 Other necessary files

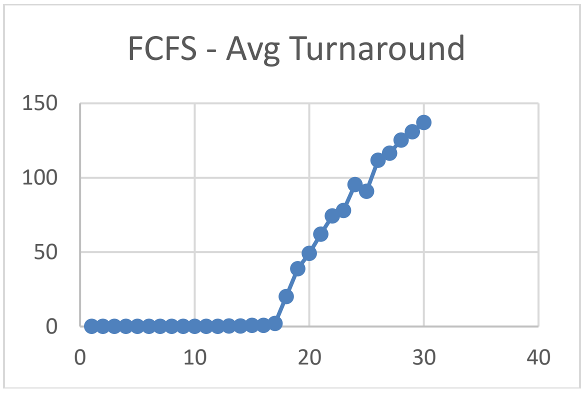
Please make sure to include all the existing csv files as they need to be overwritten. My program does not create new csv files and will simply say csv file not found if you attempt to run and save results without the file in the current folder.

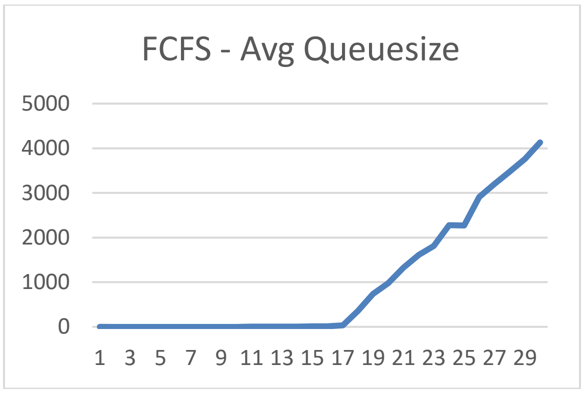
4 RESULTS

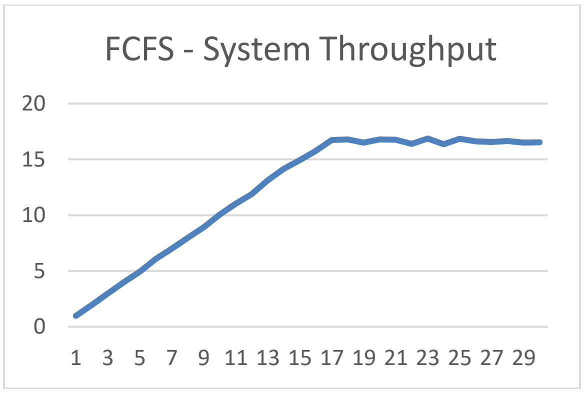
4.1 First Come First Serve

4.1.1 Data









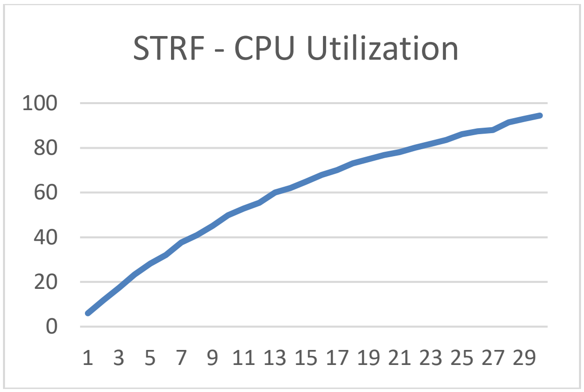
4.1.2 Data Interpretation

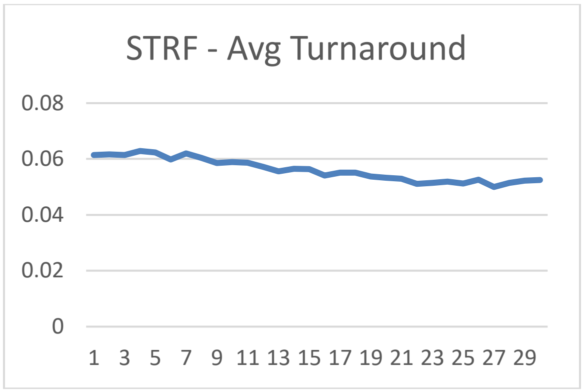
The simulator managed to successfully, record data for each of the 4 metrics specified in the project requirements. The results can be interpreted as follows:

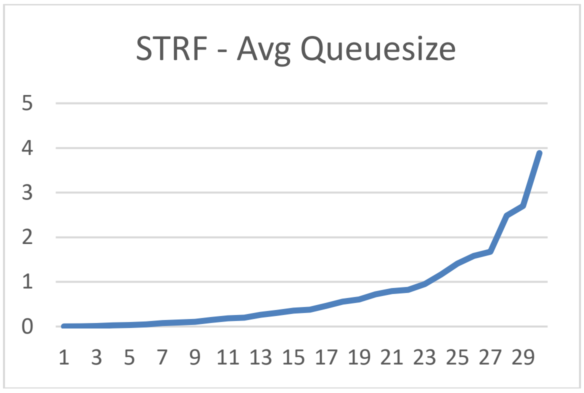
1. CPU Utilization: As the lambda value rises more processes are arriving meaning that the utilization is correctly going up. It maxes at 100% which is expected.
2. Avg Turnaround: As the lambda increases the system queues start building up and each process must wait more time to get serviced.
3. Avg Queue Size: As the lambda increases the system queues start building up since the CPU cannot handle all the events as soon as they arrive.
4. System Throughput: Throughput increases initially as more processes arrive and the CPU can handle them but once the utilization reaches its max point the throughput plateaus as it cannot handle all the events as the arrive.

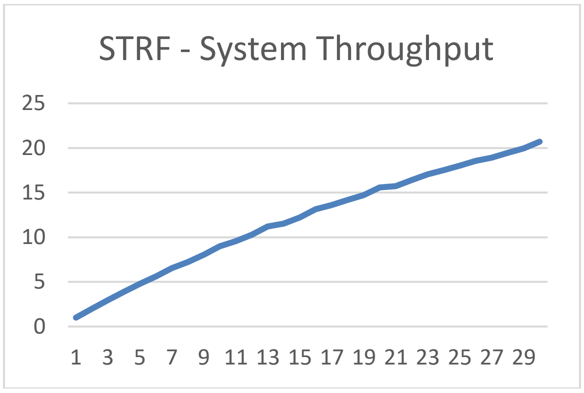
4.2 Shortest Remaining Time First

4.2.1 Data









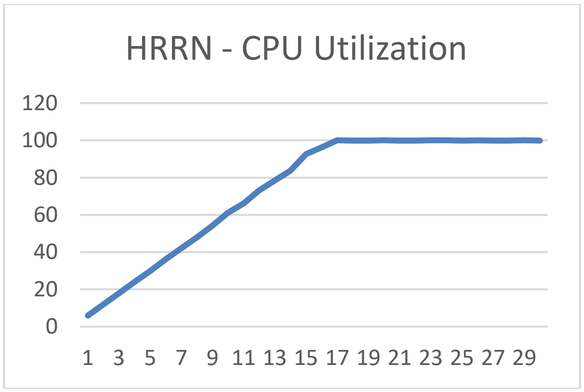
4.2.2 Data Interpretation

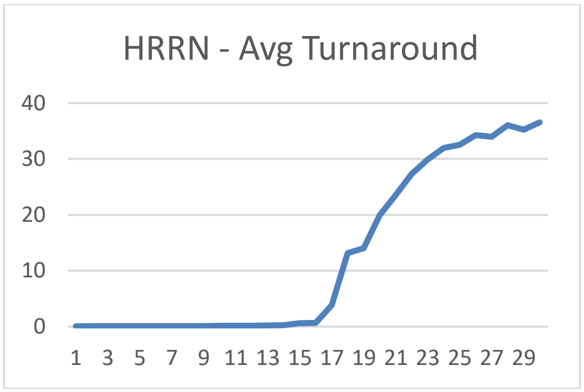
The simulator managed to successfully, record data for each of the 4 metrics specified in the project requirements. The results can be interpreted as follows:

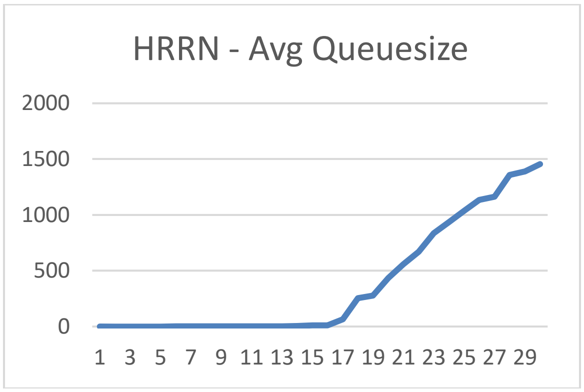
5. CPU Utilization: As the lambda value rises more processes are arriving meaning that the utilization is correctly going up. It maxes at 100% which is expected.
6. Avg Turnaround: Since SRTF suffers from starvation the turnaround time is always low because it completes short processes while ignoring the larger processes. If enough short processes arrive the larger processes never get done.
7. Avg Queue Size: As the lambda increases the system queues start building up since the CPU cannot handle all the events as soon as they arrive. The growth is more gradual than FCFS because SRTF takes care of short processes early and there is less build up.
8. System Throughput: Throughput will rise because the more processes that arrive, the more of a chance a short process is created. This means that the probability of a shorter process being created is higher with a higher lambda. Unfortunately, this means that larger processes also have less of a chance of being handled with a larger lambda.

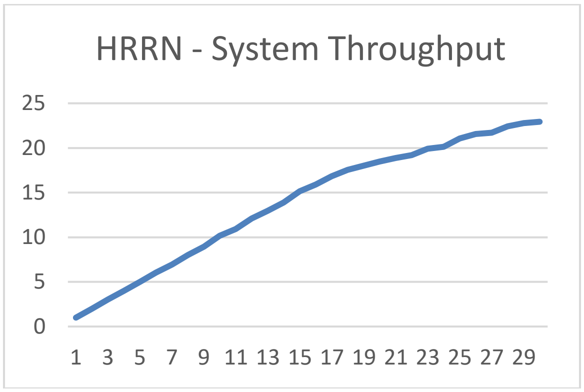
4.3 Highest Response Ration Next

4.3.1 Data









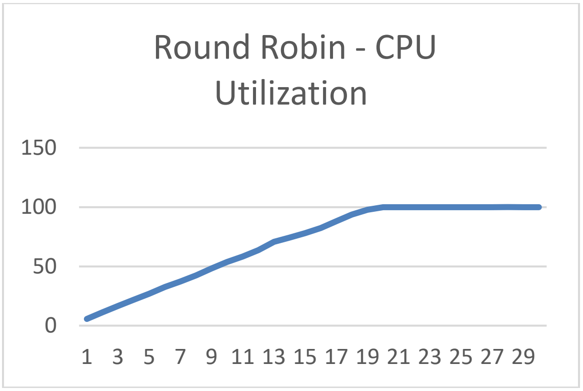
4.3.2 Data Interpretation

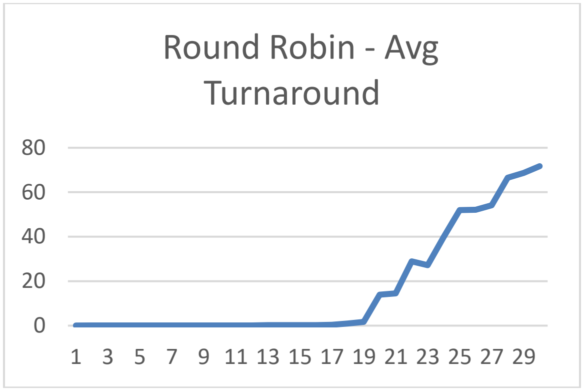
The simulator managed to successfully, record data for each of the 4 metrics specified in the project requirements. The results can be interpreted as follows:

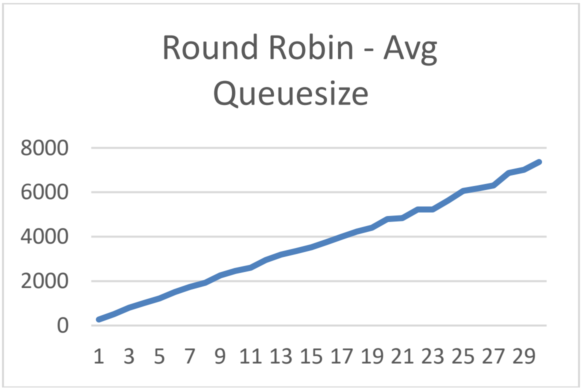
9. CPU Utilization: As the lambda value rises more processes are arriving meaning that the utilization is correctly going up. It maxes at 100% which is expected.
10. Avg Turnaround: As the lambda increases the system queues start building up and each process must wait more time to get serviced. Although it behaves like first come first serve, the avg turnaround time is lower due to HRRN considering the time processes spend in the system.
11. Avg Queue Size: As the lambda increases the system queues start building up since the CPU cannot handle all the events as soon as they arrive.
12. System Throughput: Throughput continues to increase until it eventually plateaus. HRRN performs better than FCFS since the short processes do have a small priority over larger ones initially until the larger ones spend more time in the system.

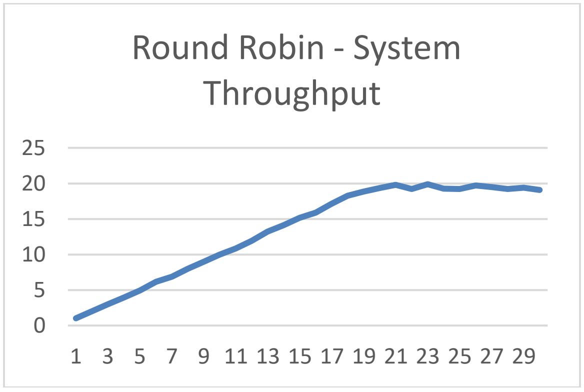
4.4 Round Robin Quantum of 0.01

4.4.1 Data









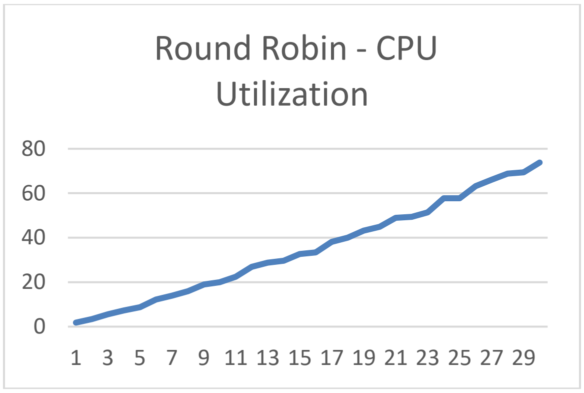
4.4.2 Data Interpretation

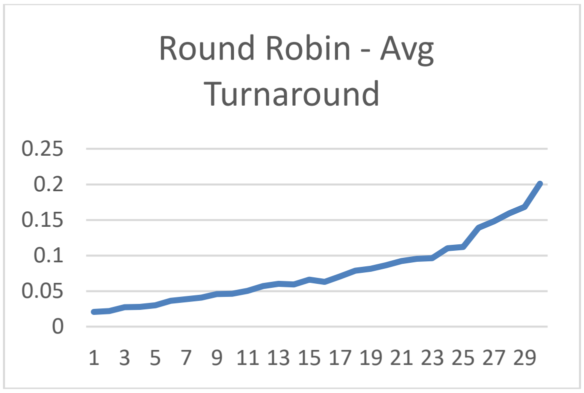
The simulator managed to successfully, record data for each of the 4 metrics specified in the project requirements. The results can be interpreted as follows:

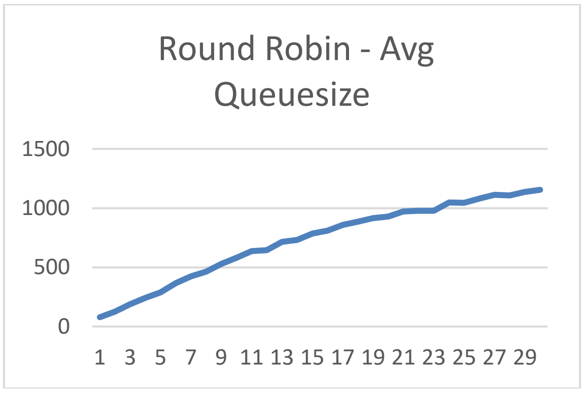
1. CPU Utilization: As the lambda value rises more processes are arriving meaning that the utilization is correctly going up. It maxes at 100% which is expected.
2. Avg Turnaround: As the lambda increases the system queues start building up and each process must wait more time to get serviced. Compared to HRRN it performs a little worse since it treats each process with equal importance.
3. Avg Queue Size: As the lambda increases the system queues start building up since the CPU cannot handle all the events as soon as they arrive.
4. System Throughput: Throughput continues to increase until it eventually plateaus. It happens earlier than HRRN since the long processes keep getting serviced and piling up on top of the short processes.

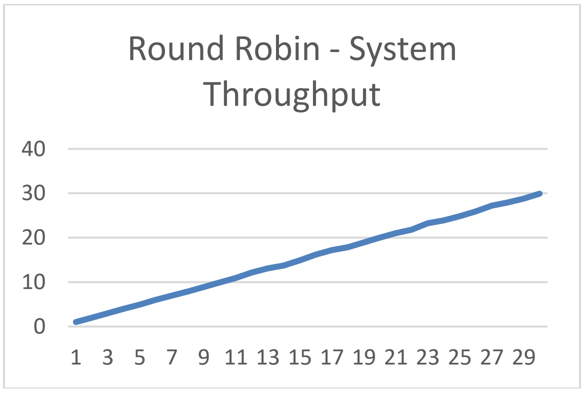
4.5 Round Robin Quantum of 0.2

4.5.1 Data









4.5.2 Data Interpretation

The simulator managed to successfully, record data for each of the 4 metrics specified in the project requirements. The results can be interpreted as follows:

5. CPU Utilization: As the lambda value rises more processes are arriving meaning that the utilization is correctly going up.
6. Avg Turnaround: these results do not match what is expected. Turnaround should be much higher. The overall shape of the curve is correct.
7. Avg Queue Size: As the lambda increases the system queues start building up since the CPU cannot handle all the events as soon as they arrive. It increases less than the previous Round Robin since the shorter processes get finished in a large quantum.
8. System Throughput: Throughput increases initially as more processes arrive. It will eventually plateau but since the quantum are larger, more of the small processes will finish inside of them and therefore create different results from the previous Round Robin.

5 FINAL OBSERVATIONS

Simulator produces accurate results. It handles incorrect input and has timeouts to prevent infinite loops. Overall quality of project is high. Major concerns are the fact that HRRN takes a long time to run. Currently I am using a priority queue to sort the processes in O(log(n)) time and lookup the top process in O(1) time. This could be sped up to O(1) time sorting with a heap data structure.

6 ACKNOWLEDGMENTS

The author wishes to acknowledge Texas State University for providing a testing grounds for all calculations as well as many years of education and hard work that gave the author the capacity to accomplish everything written down.

7 REFERENCES

Operating Systems Concepts, Silberschatz, Galvin and Gagne, 9th Edition.