Scott Hockaday

CS Fundamentals

Due: 4/11/22 @ 11:59

**CS3360 Programming Assignment 2**

Submitted Files/Programs w/descriptions:

**PA2.cpp**: This is a C++ file containing Programming Assignment 2 in the project. In this project, we are going to build a discrete-time event simulator for a First-Come First-Served (FCFS) CPU scheduling algorithm. The goal of this project is to assess the impact of different workloads on the average turnaround time of processes, total throughput (number of processes done per unit of time), average CPU utilization, and average number of processes in the Ready Queue. The simulator needs to generate a sequence of processes. For each process, we need to generate its arrival time and its requested service time. We can assume that processes arrive with an average rate 𝜆 that follows a Poisson Distribution (hence Exponentially Distributed interarrival times). The service times are generated according to an Exponential Distribution. We will vary 𝜆 to simulate different loads while keeping the average service time fixed. The simulator should stop after 10,000 processes complete, then it should output the metrics. The simulator should take 2 command-line arguments: The first is the average arrival rate and the second is the average service time. We will vary the average arrival rate, 𝜆, of processes from 10 processes per second to 30 processes per second with increment step of 1 (i.e., we are doing 21 simulation runs for 𝜆 = 10, 𝜆 = 11, 𝜆 = 12, … 𝜆 = 30, respectively). The service time is generated according to an Exponential Distribution with an average service time of 0.04 second. For each run, you would need to output the four itemized metrics above.

**README**: This is a text file containing the programming language I used for this project, the specific instructions/commands on how to log into the CS Department Linux Server (zeus.cs.txstate.edu), and how to compile and run the cpp file for the project.

**The Plots**: This is a brief description of the results and a graph of a single plot for each one of the above metrics (with 𝜆 as the x-axis). That is, each plot would have 21 data point, and there are four plots (one for each metrics). The plots would also include a description and/or discussion on the interpretation of the plots.

**Average Turnaround Time:**

Chart, line chart

Description automatically generated

**Total Throughput:**

Chart, line chart

Description automatically generated

**Average CPU Utilization:**

Chart, line chart

Description automatically generated

**Average Number of Processes in Ready Queue:**

Chart, line chart

Description automatically generated

**Plot(s) description:** All of the following plots have a lambda starting from 10 all the way to 30 with a total of 30 and the average service time is an Exponential Distribution of 0.04 seconds. According to the first plot (Avg. Turnaround Time), the plot starts pretty low at about 0.067 s then exponentially increase until about when lambda = 15. After lambda increase after 15, the turnaround time starts to decrease. According to the second plot (Total Throughput), the plot starts at when lambda = 10 and increases a total of 1 after lambda increases. When it reaches lambda = 25, it stays at 25 for the rest of simulations. According to the third plot (Avg. CPU Utilization), the plots starts at about 0.40 when lambda = 10. The plot keeps increasing linearly by 0.04 (original service time) after lambda increases by 1. According to the fourth plot (Avg. Number of Processes), the plot starts at about 0.03 when lambda = 10 then continues to increase exponentially when lambda keeps increasing.