Round Robin Project

CSCI 330-M02

Operating Systems

Kendall Molas

CPU Scheduling
CPU scheduling is a necessary for dealing with processes that constantly enter the operating system. CPU scheduling deals with processes that run on the CPU to execute its instructions. There are several different scheduling algorithms such as:

1. First Come First Serve (FCFS)
2. Shortest Job First (SJF)
3. Priority Scheduling
4. Round Robin

The goals of CPU Scheduling is as follows:

1. CPU is constantly busy.
2. Maximize throughput of the processes
3. Minimize turnaround time
4. Minimize waiting time
5. Minimize number of context switches

In this assignment, the Round Robin scheduling algorithm was focused on solely. The algorithm of Round Robin goes as follows, processes will enter the operating system one by one and be put into the ready queue to wait for CPU to be available. When CPU is available, the process will run for some amount of time. This time that the process will run on the CPU is called the time quantum.

Implementation
For this implementation of Round Robin, I used C. There are three many structures: Queue, CPU, and Process. The queue structure contains the array size, front item, rear item, original size, and its current size. This queue would be filled after read csv method is executed. The CPU structure contains information that would be used for displaying the final results of the algorithm at the end. Some of the information includes total waiting time and total context switches. Lastly, the process structure contains information of the arrival time, burst time, turnaround time, etc.

After the queue is filled with the processes read from the csv file, it would go into execution. It would constantly loop through the queue as long as the front process is not null. During the loop cycles, it would subtract the burst time from the time quantum that the user will input. After the difference between the burst time and the time quantum is computed, it would check if the burst time is zero. If it is zero, it would compute other information such as the waiting time and turnaround time. Information such as the waiting time and turnaround time are then passed to the CPU variables for later analysis. If burst time is not zero, the process' context switch would then be incremented by one. The process would then be dequeued, and it would repeat the process.

Analysis of Output Data
In regards to the actual output data, the CPU was constantly busy as shown in the screenshots. Other information was outputted as well including the total time it took to run based off the time quantum, the total idle time, average waiting time, total throughput, and the total amount of context switches. These times were determined based off the system clock which I included through the header time.h.