Project 1: CPU Scheduling Algorithm Simulations

CS 431 Operating Systems
Brandon Palomino
13 October 2017

Introduction

For this project I was assigned to not only implement the fundamental CPU Scheduling algorithms but also thoroughly understand their natures and purpose for which they were designed.

Analysis

I started by learning how the scheduling algorithms came to be through the progressing development of computers, which led to a need of advanced scheduling which could handle numerous processes. Followed by learning of the three different environments which they would be used for. Those environments being: batch, interactive, and real time systems.

For this project, the algorithms used only covered the scope of batch and interactive systems. Those algorithms being specifically: First Come First Serve, Shortest Job First, Round Robin, and Lottery scheduling. We were supplemented with test data files which had a set of processes with burst times and priority values. I implemented the algorithms and setup my program to allow the input of a file at runtime which would load the data and simulate all the schedulers one by one. Once all the processes had been completely finished the scheduler would finish and output all the processes along with its burst time, wait time, turnaround time, average turnaround time, and average wait time in order to promptly show what had happened throughout the scheduler's simulation. An important note to take into account was that these scheduler implementations were set so that the scheduler takes 3 units of CPU time in order to switch processes. This does have an effect on how it long it take the scheduler to each job and on the output results.

What can be observed from my output, is that each scheduling algorithm did a slightly better job than the previous one in final turnaround time as well as the averages. However there are exceptions with the batch systems algorithms doing better than round robin in their averages for turnaround and wait times. The lottery scheduler overall was the best algorithm out of the four. This was expected of lottery because of its use of prioritizing the processes and giving more cpu time to the processes which had a higher priority. When the more processes were added in test data 3 and 4, it was apparent how the interactive system algorithms Round Robin and Lottery fared much better at handling all the processes and keeping the total turnaround time at about half of the batch systems schedulers.

Conclusion

With all that being said, CPU scheduling holds, truly an important job in all operating systems. As os and computers continue developing so will the scheduling algorithms.