CS 4348.004 Operating Systems Concepts Max Lichter
Spring 2024
Summary
Due: 4/27/24

Overview:
The code that I produced is a scheduler for scheduling a set of jobs using multiple different schedule algorithms. For the project, we used the first-come-first-serve and round robin algorithms. The main objectives that this project offers is to offer users that use it the capability to execute different scheduling methods on an inputted job set and visualized the resultant job execution sequence through a hashmap / gantt chart. This presentation gives an scalable platform for job management and total execution analysis

Difficulties Encountered:
I encountered a ton of difficulties when developing a scheduler capable of effectively handling such a varied job set. Executing MANY scheduling algorithms gave me several challenges. Here are a couple of the challenges. The first challenge that plagued the production of this project was concurrency management. Making sure that there was proper synchronization and thread management, ESPECIALLY in the Round Robin algorithm, where tasks are processed concurrently was an extremely large major challenge. I was stuck on the Round Robin algorithm for hours and hours not knowing what to do. Another challenge that I faced was task queue handling. Managing the task queues and optimizing the data structures for efficient processing of the arrival, particularly in certain scenarios with varying arrival rates and job lengths was such a pain to deal with. I am not terribly efficient with data structures to begin with so this posed a major challenge. The final challenge that I will talk about is actually just resource optimization. Balancing resource allocation and minimizing the job wait times in the middle of fluctuating workload conditions required very complex strategies for task scheduling and execution that I was not prepared to handle from the beginning. Overall the challenges that I faced through the course of making this program definitely made me more proud of the final product when I achieved it. I am happy to say the difficulties I encountered are no more

Lessons Learned:
This will go hand and hand with the difficulties encountered section of this summary. To start, I learned a plethora of information that I didn't know about before making this project. Firstly, I learned a ton more concurrent programming principles. Although I learned a TON of concurrent programming principles in the last program, this project really opened a new eye for me in terms of concurrent programming. To understand and implement concurrent programming concepts, not limited to synchronization mechanisms and thread interactions was hard work, but in the end, I learned how to manage both of those behemoth tasks. I also learned how to better utilize data structures. Using appropriate data structures such as queues for efficient task management and processing facilitated a relatively smooth job execution for the program. Finally, I learned the algorithms themself and how to efficiently optimize them. I enhanced the two scheduling algorithms to improve efficiency and throughput while also ensuring that there was total resource utilization which in turn, minimized latency

Results:
In the end, the scheduler that I made showcased the successful running of the first-come-first-serve and round robin algorithms on a ton of different ranges of input job sets. I believe that my program provides a very comprehensive visualization of job scheduling and

execution sequences, which also gives very prestigious insight into different strategies of scheduling themself. Overall this project really does demonstrate the effectiveness and adaptability of the implemented algorithms in managing different input job.txt files.