***Progress Report CSCI 432 Final Project***

**Authors: Devan W. Eastman-Pittam, Ben F. Miller**

**Introduction:**

Our project that we are working on for the Final, is that we are cross comparing three separate task scheduling algorithms. These are namely Shortest Remaining Time, Round Robin, and Priority Based schedulers. The metrics we are examining are total time taken, and time taken between individual algorithms to complete the same tasks whether they are short, long, or priority. In this report we will go over our current progress on the project, Immediate next steps, and our course of action such that we can have the most optimal steps taken to finish the project by the deadline.

**Current Progress:**

The majority of the work accomplished has been on researching the CPU scheduling algorithms and comparing/contrasting them. We have a shared document in which we have written out details about each of the three algorithms chosen. This document has served as the root of all concrete results as of yet. It features an outline for the final paper and a references section. Future progress will be largely based around the refinement and expansion of this document.

In addition to the written work accomplished so far, we have a significant amount of progress done on the code, mainly in java. We think that some statistics on the algorithms would be beneficial to the video and write up and that they are an interesting, practical way to analyse the algorithms. We have also implemented a task generator that we will use to build and analyze task lists of any size. We have chosen to implement all three algorithms to better compare them. Because they are all scheduling algorithms, each implementation is largely based on the priority queue implementation. With the wonders of object oriented design, we were able to encapsulate most of the code in common and significantly reduce the complexity of the code.

The simulation is very simplified compared to an actual cpu scheduler and uses a simpler priority system with a priority of 1 being the highest. The task generator is easily adapted to any number of priority tiers. The schedulers also know the exact amount of computation time required for each task. This is a luxury not found in most operating system level scheduling settings, but is more common in places like web servers.

**Immediate Next Steps:**

Our immediate next steps will be to finish the rest of our code which is already nearing completion and flesh out our information we have on the algorithms before doing the video. Some more things which could be added would be, more pros and cons for each algorithm, possibly adding the history of these types of algorithms which could include creators and significant contributors, and fleshing out our summaries of them.

For the specific order it will be, “Finishing the Code”, “Fleshing out the details”, “Recording the video”, and lastly “Performing Edits and Revisions”. Finishing the code is pretty self explanatory and includes writing and revising code such that it performs correctly and as expected.

Fleshing out the details will include increasing the total amount of information and diversifying what is currently written, and then revising it into the best format for the end project. Recording the video is self explanatory and will include recording and revising the video till the desired product is achieved. And finally Edits and Revisions will be to go back over the whole project and add any missed items and to fix any bugs or errors.

The way which we will complete this is that we have self scheduled between ourselves for a couple of meetings to finish that which has not currently been completed, and we will continue to work and revise so that we don't have anything which will pop up at the very last minute which could disrupt our plans.

**Link to Our Actual Document For The Project:**

[**https://docs.google.com/document/d/1gmQ7BPS0tS3bOsaURWTfM\_YA8q-ojYqwPk3F39zOAQ8/edit**](https://docs.google.com/document/d/1gmQ7BPS0tS3bOsaURWTfM_YA8q-ojYqwPk3F39zOAQ8/edit)