Abdullah Almaroof and Julio Anacleto

TCES 430 - Autumn 2025

November 22, 2025,

Project 3

Mutual Exclusion: For mutual exclusion we learned throughout the program, that literally everything needed to be locked if the other threads had the possibility of using it at the same time. In order to do this, we had to make five locks, one for each queue, one for the global variable that would tell each job what it's unique ID was, and the last one was used to increment a global variable that would keep track of jobs that were completed.

Challenges Faced: Our primary challenges was figuring out how to make the three functions (cpu_worker_thread, io_worker_thread, and job_submitter_thread) would be created and used, this challenge was because we had to make sure that when locked something, we didn't forget to unlock it, this was a problem because deadlocking kept on occurring, but once we figured it out it was great. Another problem was figuring out how to initialize randomly the phases of each job and how much time those phases took. Also, another challenge was figuring out how to implement queues/scheduler with the threads, it took using a lot of pointers and three different queues.

Contributions:

- Abdullah – Created the program, created report
- Julio – Helped debug the lock issue, made the makefile, helped with report