Kurt Slagle

Samuel Sudekum

Project 2 – Report

Implementation Issues

- Segmentation faults with the queue data structure
- Queue removal oversight from **scheduler_job_finished(…)**
- Ridiculous programming (nooby) errors
- Issues managing the implementation of multiple cores
- Handling of void pointers in C, since C++ does this differently
  - Ended up being simple, since C allows casting of the pointer
- **priqueue_remove(…)**
  - If a structure is sent to it, there can be so many attributes inside the structure
    - Using the unknown data type (void *) and the comparison function is difficult when implementing the removal of multiple instances of the same structure
- Management of two separate code bases
- Debugging issues in scheduling algorithms with PSJF and multiple cores
- Calculating the time remaining was very problematic
- Calculating the response time was also problematic
- Round Robin presented issues with offering the job to the queue at the appropriate time

Solutions To Problems

- Segmentation faults – debugger used to locate the locations of segmentation faults
- Oversight resolved from the **scheduler_job_finished(…)** function
- Increased careful scanning of syntax to avoid simple programming

mistakes

- Like stated above, somewhat easily handled due to the casting ability
- Small tweaks in code to calculate the time remaining were necessary to get correct values
- Used GitHub to manage code

Feedback

- Having a code base to begin from was very helpful
  - Instead of spending so much time figuring out how to structure a complex program like this, we were able to allocate more time to working out the implementation of the requirements
- Having very-well-documented code was very helpful
  - Knowing exactly what each function needed to do, exactly what information would be given, and exactly what needed to be returned removed a lot of confusion about implementation