Samuel Wynsma, William Tadlock, Quinlan Reade

System Design Documents

System Requirements:

We are making a dispatcher simulation. Our UI will be a classroom simulation, with the students as processes or our model entities. Students come into a classroom and go out of the classroom. The classrooms are the files that will have the students stored in them. All data will be tracked and displayed for every student that enters and leaves a certain classroom. When a student leaves, the process associated with them ends. The discrete events are: a student enters a room, and a student leaves a room.

PCB Structure: We want a unique PID for every process/student that is in a queue or in a classroom at any given time. Once a process/student exits the classroom, that PID is then recycled and then can be reassigned to any process that will then be added to any of the queues after some reassignment algorithm.

Process state:   
Scheduling algorithms for any given queue. The registers store the information for any given process, classroom, and queues.  
The PCB will need to have an arrival time for the last queue the process arrived in, and the last classroom that it arrived in.

Queues: There will be anywhere from 1 to N queues, and there will be N classrooms. When a student is in a classroom, something will be sent to a file as the student leaves the classroom. Each classroom connects to its own file. As a student exits a classroom, they can get back in line in the shortest queue. Each queue will be running an FCFS algorithm. The queue that stores the queues is an HRRN algorithm. The queue at the front of the HRRN algorithm at any given point is determined by the length of time the process at the front has been at the front of that queue. If a queue gets over X length, then a UI prompt will pop up and the user can manually modify which queue gets taken from first and reorganize all queues.

Scheduling algorithms: We are doing a multilevel feedback scheduling algorithm. Each queue can have a different algorithm specifically. There will be a master process and a UI for any queue enters. When a process is ready to be placed in a queue, it will go through the master process. This will determine if it goes to the UI or if it is immediately placed in a queue.  
FCFS will probably be the main queue for entering a classroom. The scheduling algorithm for the queues specifically will be an HRRN scheduling algorithm.

Project Management Plan:

William will handle anything regarding UI. Sam will handle anything involving multithreading, as well as some of the backend coding for the assignment. Quinlan will do the majority of the testing for this project.

Sam is responsible for all queues and the queue scheduling algorithms and anything involving multithreading.

William will create the main UI and anything involving data inputs and outputs. He will also do queue work cleanup, meaning anything that Sam does not complete.

Quinlan will mainly focus on testing, however, anything not covered by William and Sam will be covered by Quinlan. Quinlan will handle all context switching in the simulation.