

Lab 5: Process Scheduling

Mason Walls

Section 7

UID: 910333448

During this lab I learned about 4 different process scheduling techniques. This lab taught me how operating systems schedule processes and how context switching can be implemented. The first algorithm I wrote was first come first serve. This required me to wait for a process to be ready, and run that one until it ended. I encountered my first problem in this part. I didn't understand right away why we needed the current time as an input to the function. I then realized that arrival time must be equal to or less than current time to consider it ready. I then wrote shortest remaining time. This required the same looping as first come first serve, but instead of arrival time, I checked remaining time. For round robin, I continued to use the while loop to find the first available process. I had to use a static variable called last_pid for this part. If last_pid was -1, I knew that no process had started yet, so I looked for the first ready one. Once last_pid was a value, I searched from last_pid to NUM_PROCESSES for the next available process where pid isn't equal to last_pid. If none were found, I looked from 0 to last_pid. If none were found again, I checked if last_pid was still running and returned it. Otherwise I returned -1. For round robin priority, I used the same process as round robin, but three different times for the priority levels. If no processes, I returned -1. If there was a process ready and last_pid was -1, I returned the one with the highest, and set last_pid to it. Then I used the same loops as before, but split it into 3 based on priority. I looked for the next high priority process. If none found, I checked if last_pid was high priority and still running, then returned it or -1. I did this for each priority level.

Implementing these algorithms required some thought. I was very surprised to see that first come first serve had the shortest average turn around time. Seeing these different ways to schedule showed me that the type and length of a process will matter when we decide how to schedule them.