Process Scheduling

Name - Kunal Sharma
Roll Number - 2021331
Branch - CSD

Code Description

In the main for loop, I first create three child processes that use the excel() command to run the three separate bash files (bash1.sh, bash2.sh, and bash3.sh). The actions that take place in each bash file are to copy the kernel, navigate to the copied kernel directory, and then compile the copied kernel using the make command. Using sched_setscheduler(), I set up a process policy, which is used as an attribute by the sched_setscheduler, and each child process uses param.sched priority to set the priority of the current process (). Now I'm starting a for loop statement with the requirement that all of the child processes must be finished. I'm using waitpid() in this while loop to wait for the processes with their appropriate process IDs to complete. Finally, I'm including a while loop to figure out the time that clock_gettime() marked using the attribute CLOCK_REALTIME.

The output of the code shows a tendency for SCHED_RR to execute in the shortest amount of time and SCHED_FIFO to execute in a slightly longer amount of time because SCHED_FIFO performs poorly when the same task is being performed by these two processes (given priority of SCHED_RR and SCHED_FIFO is same) . SCHED_OTHER takes a lot longer than SCHED_FIFO because SCHED OTHER has the lowest priority by default and is executed last.