READ\_ME Q2 OS ASS2

The procedure of this question has been divided into two c programs: ‘ass’ and ‘count’. The ‘count’ program has used clock\_gettime function to store the starting time in a variable. Then, a for loop runs implementing counting to start from 1 to 2^32. This is followed by storing the end time in another variable. The difference of these times gives the duration taken to complete the counting. This is stored in a file named “output.txt” using file i/o. Error handling has been done for file i/o.

The ‘ass’ program is the program through which we run and prioritize the different processes SCHED OTHER, SCHED RR, SCHED FIFO. The code contains a for loop that runs 3 times. This loop ensures that all the processes are running (**and has no relation to prioritizing the processes)**. For each process, a child process is created using fork(). Then, values for the priority are set and run in the nice(). Post this, the count program using exec() is called for each of the three different processes. At last, all the processes are terminated using waitpid(). Error handling has been done for all system calls.

The python program takes the execution times present in the file “output.txt” and then plots these on the x-axis. The y-axis of the plot has the frequency of process call. The labels and the execution times show the priority of each process decided by the system.

Explaining the outputs:

We can see that the processes have been assigned different priorities by the system. The process with higher priority is run first and hence, its time taken is less and the process with the least priority is run at the end. This can be seen form the graphs constructed as well.