LAB-4 readme

-Mayank Raj(B19CSE053)

Implementation details:

- Generation of different processes using generator.c and process.c using POSIX and xterm.
- Implementation of Round robin and Priority round robin in shed.c
- Queue management of two types -priority and normal queue.

Features:

- Generator file able to generate processes in xterm.
- Process file is running correctly displaying states of each process.
- Scheduler file is correctly running and implementing both the algorithms.

Steps to run:

****Note: Do in the order given and the linux environment is necessary and install xterm if not installed.(sudo apt install xterm)**

- Open terminal and write the following code.
- gcc process.c -o process
- gcc generator.c -o gene
- ./gene
- .Open another terminal and write:
- gcc shed.c -o shed
- ./shed <no. of processes> <algorithm code> <time quantum>
  These are command line inputs. Algorithm code => 0 for robin round and 1 for Priority robin round
- Eg=> gcc shed.c -o shed;./shed 3 0 1

- In first terminal After running ./gene enter inputs as <no of Inputs> <priority> <probability> <start-time>
- Eg. 3 2 0.5 1
- For adding more processes repeat the previous step.

Bugs:

- Maximum size for priority queue(max heap) implemented is 1000 and when no. of processes increases then it can throw an error.
- When no. of processes is large then low priority processes can take a long time to execute and could not get executed at all.
- Probability for I/O should be less than 1 and greater than 0 and should be not more than 18 decimal places.

Output files:

- For each process output<id>.txt file generated containing and the different timeline of its execution. Containing id ,response time waiting time and turnaround time.