

160CS093

The LNM Institute of Information Technology
Operating Systems Lab (CSE223)
Mid-Term Examination (Lab) Odd Semester 2017-18

Lab Session-2
SET-A

Note:- No internet connectivity/usage is permitted. All help is readily available on the system itself. Copy/cheating will get you straight a F grade!
The 'Test Example' given is just for your help to verify the program. Do not hard-code it during implementation.

Write a C program to simulate a multi-threaded preemptive priority scheduling algorithm based on dynamically changing priorities. When a process is waiting for the CPU (in the ready queue, but not running), its priority increases by 'a'. When it is running, its priority increases by 'b'. All processes are given a priority of 0 when they enter the ready queue. The main thread will take arrival time and burst time for six processes as user input. Then the main thread will create a new thread named 'scheduler'. Scheduler thread is responsible for implementing the scheduling algorithm. Following are the rules:

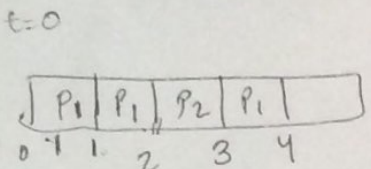
1. The higher number indicates a higher relative priority.
 2. The process with the highest priority gets the CPU first.
 3. Processes are preempted when higher priority process arrives.
 4. At any point if processes have same priority, select the one which comes first
 5. The value of 'a' is 3 and 'b' is 1.
 6. Scheduler will check for the priority after each time unit lapse.
- In the end, the main thread will display a chart with the following:

- The order of process completion
- Turnaround Time
- Waiting Time
- CPU Utilization Rate (Assume context switching will take no time)

Test Example: Assume, following are the user inputs

Priority
 $P_1 \rightarrow 2+3=5+1=6$
 $P_2 \rightarrow 3+4=7$
 $P_3 \rightarrow 3$
 P_4
 P_5
 P_6

Process	Burst Time	Arrival Time
P ₁	5	0
P ₂	4	2
P ₃	2	3
P ₄	3	6
P ₅	1	10
P ₆	4	15



Output: The program should display the following:

Process	Turnaround Time	Waiting Time
P1	6	1
P3	7	5
P2	9	5
P4	8	5
P5	5	4
P6	4	0

100%