

**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:

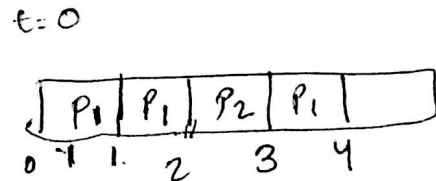
1. The higher number indicates a higher relative priority.
2. The process with the highest priority gets the CPU first.
3. Processes are preempted when higher priority process arrives.
4. At any point if processes have same priority, select the one which comes first.
5. The value of 'a' is 3 and 'b' 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

Process	Burst Time	Arrival Time
P <sub>1</sub>	③ 5	0
P <sub>2</sub>	③ 4	2
P <sub>3</sub>	2	3
P <sub>4</sub>	3	6
P <sub>5</sub>	1	10
P <sub>6</sub>	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%

**Lab Session-1**  
**SET-B**

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 Priority based Round Robin scheduling algorithm. The main thread will take arrival time, burst time and priority 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 length of a time quantum is 8 units.
3. Processes are only preempted via the time slice of the Round Robin
4. The process with the highest priority gets the next time slice.

In the end, the main thread will display a chart with the following:

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

Test Example: Assume, following are the user inputs

Process	Priority	Burst Time	Arrival Time
P <sub>1</sub>	40	20	0
P <sub>2</sub>	30	25	25
P <sub>3</sub>	30	25	30
P <sub>4</sub>	35	15	60
P <sub>5</sub>	5	10	100
P <sub>6</sub>	10	10	105

Output: The program should display the following:

Process	Completion Time	Response Time
P <sub>1</sub>	20	0
P <sub>2</sub>	66	0
P <sub>4</sub>	81	6
P <sub>3</sub>	90	3
P <sub>6</sub>	118	3
P <sub>5</sub>	120	0