

**Q2A:**

In round robin scheduling, a time quantum is initialized. Processes enter the queue according to their arrival times. The first process starts execution, and halts if its burst time is greater than the time quantum. The halted process is sent at the back of the queue. This is repeated for all the processes. When the halted processes resume execution, their remaining burst time is completed if it is smaller than the time quantum, or else, they are halted again to be sent at the back of the queue. These steps are repeated until all the processes get executed.

In modified round robin scheduling, the time quantum is calculated rather than being arbitrarily selected. Its process is as follows:

Mean and median of the burst times are calculated. If the mean > median, then time quantum is  $(\text{mean} * \text{max burst time})^{0.5}$ . Else, it is  $(\text{median} * \text{max burst time})^{0.5}$

Rest of the procedure remains the same as round robin scheduling.

This modification helps in reducing the average waiting and turn around times, as observed in the implementation.

```
shubhdhebar@ububtu-hp:~/Documents/os/endsem_practical$ gcc rr_scheduling_2A.c
shubhdhebar@ububtu-hp:~/Documents/os/endsem_practical$ ./a.out
Processes  Burst   WaitingTime  TurnAroundTime
1          7       28            35
2          5        5            10
3         12       27            39
4          8       32            40
5          9       33            42
6          5       24            29
Average waiting time = 24.833334
Average turn around time = 32.500000
shubhdhebar@ububtu-hp:~/Documents/os/endsem_practical$ gcc mod_rr_scheduling_2A.c -lm
shubhdhebar@ububtu-hp:~/Documents/os/endsem_practical$ ./a.out
Modified time quantum = 7
Processes  Burst   WaitingTime  TurnAroundTime
1          7        0             7
2          5        6            11
3         12       29            41
4          8       33            41
5          9       33            42
6          5       28            33
Average waiting time = 21.500000
Average turn around time = 29.166666
```

**Q3:**

In the Producer-Consumer problem, a producer creates an item to place it in a buffer, and the consumer pick items from it to consume. It is to be ensured that these events do not occur simultaneously, i.e. a consumer cannot consume an item while a producer is placing one in the buffer.

To do so, semaphores are initialized, to keep track items in the buffer.

Buffers: Full, Empty and Mutex

As the producer creates an item;  $\text{Empty} \rightarrow \text{Empty}-1$ ;  $\text{Mutex} \rightarrow \text{Mutex}-1$ ;  $\text{Full} \rightarrow \text{Full}+1$

As the consumer consumed an item;  $\text{Full} \rightarrow \text{Full}-1$ ;  $\text{Mutex} \rightarrow \text{Mutex}+1$ ;  $\text{Empty} \rightarrow \text{Empty}+1$