

Question no 01:

Reason:

For a system with 4 cores and a 1-to-1 threading model, I create 4 threads - one for each core - to maximize efficiency and for utilizing all resources efficiently.

Question no 02:

Answer:

With 8 cores available and using a 1 to 1 threading model, it's optimal to create 8 threads for the task.

Reasons:

→ Max CPU Utilization:

Utilizing all 8 cores with individual threads ensures that each core is fully engaged, maximizing computational throughput.

### → Efficient Resource Utilization:-

Since, each thread is can run independently on its core, there minimal overhead from context switching, ensuring efficient use of CPU resources.

### → Balanced Workload:-

By distributing the workload across 8 threads, the task of determining number of frequency of specific number in the array can be performed concurrently and efficiently.

→ Therefore, for a system with 8 cores and a 1-to-1 threading model, using 8 threads ensures performance and efficiently utilizing available computational resources. \*

Question No. 03:- \*

Answer:

→ I create two threads because the number of cores in system are 2 and its the optimal condition for a system with two cores as it maximize parallelism and ensure all available processing power is used.

Reasons:

With 2 core available and 1-to-1 threading model;

Optimal thread count is 2.

Each core handle one thread efficiently.

Minimizes context switching overhead.

Maintains efficient resource utilization.

Balances workload distribution effectively.

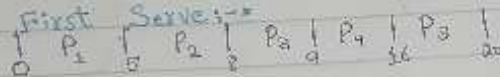
= Hassani Zaid Jaden

22PWC052144

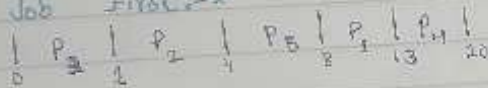
→ Answer # 4:-

(a) Gantt Chart:-

→ First Come, First Serve:-\*



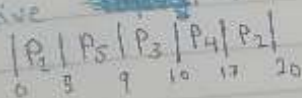
→ Shortest Job First:-\*



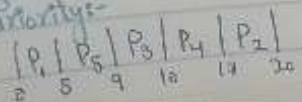
→ Shortest Remaining Time:-\*



→ Non-Preemptive



→ Preemptive Priority:-





TAT of Each Process:-  
 → First Come, First Serve

Process	Burst Time	Waiting Time	TAT = WT + ET
P <sub>1</sub>	5	0	5
P <sub>2</sub>	3	5	8
P <sub>3</sub>	1	8	9
P <sub>4</sub>	7	9	16
P <sub>5</sub>	4	16	20
		7.6	11.6

→ Shortest Job First:

	BT	WT	TAT
P <sub>3</sub>	1	0	1
P <sub>2</sub>	3	1	4
P <sub>5</sub>	4	4	8
P <sub>1</sub>	5	9	13
P <sub>4</sub>	7	13	20
		8.2	9.2

Shortest Remaining Time:-

	BT	WT	TAT
P <sub>3</sub>	1	0	0
P <sub>2</sub>	3	1	5
P <sub>3</sub>	4	4	9
P <sub>1</sub>	5	8	10
P <sub>4</sub>	7	13	17
		5.2	8.2

Preemptive Property:-

	Priority	TAT	WT
P <sub>1</sub>	4	5	0
P <sub>5</sub>	3	9	5
P <sub>3</sub>	2	10	9
P <sub>4</sub>	2	17	10
P <sub>2</sub>	1	20	17
		12.2	8.2

### CPU Utilization Rate:-

As we know that C.U.R is Total Required time by total time taken into 100,

$$C.U.R = \frac{\text{Total Required Time}}{\text{Total Time Taken}} \times 100.$$

(i) Shortest Job First:-

$$20/20 \times 100 \Rightarrow 100\%$$

(ii) Shortest Remaining Time:-

$$20/20 \times 100\% \Rightarrow 100\%$$

(iii) Non-Preemptive Priority:-

$$20/20 \times 100\% \Rightarrow 100\%$$

(iv) First Come First Serve:-

$$20/20 \times 100\% \Rightarrow 100\%$$

(v) Preemptive Priority:-

$$20/20 \times 100\% \Rightarrow 100\%$$

Waiting Time for Each Process:-

The waiting time for each process has already found in part b.

Best Performance:-

SJF and SRT have the best performance because of they have less average TAT and WT than the other one.

Round Robin SA:-

$P_1$  |  $P_2$  |  $P_3$  |  $P_4$  |  $P_5$  |  $P_1$  |  $P_2$  |  $P_4$  |  $P_3$  |  $P_1$  |  $P_4$   
 0   2   4   5   7   9   11   12   14   16   17   20

→ T.A.T and W.T:-	TAT	W.T
$P_1$	$47 - 10 = 37$	12
$P_2$	$12 - 2 = 10$	9
$P_3$	$5 - 4 = 1$	4
$P_4$	$20 - 3 = 17$	13
$P_5$	$16 - 7 = 9$	12



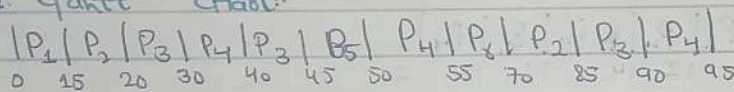
→ CPU Utilization Rate:-

$$20/20 \times 100\% \Rightarrow 100\%$$

→ Question No 05:-

→: ANSWER:

(a): Gantt Chart:-



(b): Turn Around Time:

Process	Priority	BT	TAT	W.T. = T.AT - BT.
P <sub>1</sub>	8	15	15	0
P <sub>2</sub>	3	20	85	65
P <sub>3</sub>	4	20	70	50
P <sub>4</sub>	4	20	70	50
P <sub>5</sub>	5	5	5	0
P <sub>6</sub>	9	15	15	0

Waiting Time:

Process	Priority	BT	TAT	WT = TAT - BT
P <sub>1</sub>	8	15	15	0
P <sub>2</sub>	3	20	85	65
P <sub>3</sub>	4	20	70	50
P <sub>4</sub>	4	20	70	50
P <sub>5</sub>	5	5	5	0
P <sub>6</sub>	5	15	15	0