

Date.....

Name → Suyash Jain

Course → BTech CSE

Roll No. → 2301010249

## Operating System Assignment 01

Q1.

Ans. → Even with advanced hardware, OS is essential for resource management (CPU, memory, I/O). It provides abstraction, multitasking, security and ensure programs can run safely and efficiency.

Q2.

Ans. → A real-time embedded OS is best because the device needs quick and predictable response, low power usage, and small footprint.

Q3.

Ans. → Avoid microkernel, as frequent inter-process communication adds overhead. Monolithic or layered gives better raw performance.

Q4. →

Ans. → Yes, Structure impacts performance, maintainability, and fault isolation. For example, microkernel is secure but slower, monolithic is faster but less reliable.

Q5.

Ans. → If PCB stores registers, PC and state → errors show uninitialized values.

(ii) Context switch = saving current process state, updating PCB, loading new state.

(iii) Mid-execution I/O allocation usually needs non-blocking call so process continues running.

### Part - B

Que 6.

Ans a) Total context switching time,

Save state = 2ms

Load state = 3ms

Scheduler overhead = 1ms

Total time =  $2+3+1 = 6\text{ms}$ .

b)

\* Context switching is pure overhead (no useful work is done during this time).

\* Higher switching time reduces CPU efficiency, as more time is spent switching than executing process.

Que 7

Ans. Execution time estimate:

In ideal conditions (perfect parallelism, no overhead):

$$T_{\text{multi}} = \frac{T_{\text{single}}}{n} = \frac{40}{n} \text{ second.}$$



Date.....

Single-thread = 40 sec.

With 2 threads under ideal cond<sup>n</sup> =  $\frac{40}{2} = 20$  sec.

- \* Multithreading allows overlapping of I/O and Computation, utilizing multiple cores and reducing waiting time, though in practice overhead reduces perfect gains.

Que 8.

Process : P<sub>1</sub> P<sub>2</sub> P<sub>3</sub> P<sub>4</sub>  
Burst time : 5 3 8 6

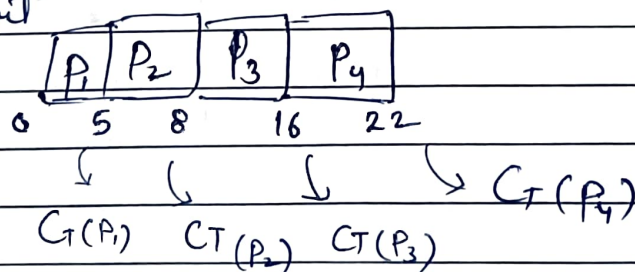
a) FCFS

Process	Arrival Time (AT)	Burst Time (BT)	Completion Time (CT)	Waiting Time (WT)	TAT
P <sub>1</sub>	0	5	5	5-5=0	5-0=5
P <sub>2</sub>	0	3	8	8-3=5	8-0=8
P <sub>3</sub>	0	8	16	16-8=8	16-0=16
P <sub>4</sub>	0	6	22	22-6=16	22-0=22

$$WT = \text{Turnaround} - \text{Burst} \quad (TAT - BT)$$

$$TAT = \text{Completion} - \text{Arrival} \quad (CT - AT)$$

Gantt Chart



Date.....

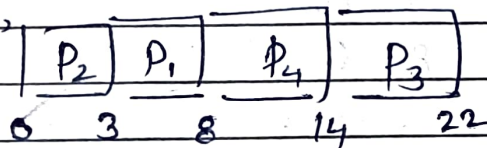
$$\text{Avg waiting time} = (0 + 5 + 8 + 16) / 4 = 7.25 \text{ ms}$$

$$\text{Avg turnaround time} = (5 + 8 + 16 + 22) / 4 = 12.75 \text{ ms}$$

(b) Non-preemptive STF

Process	AT	BT	CT	WT	TAT
P <sub>1</sub>	0	5	8	8 - 5 = 3	8 = 8 - 0
P <sub>2</sub>	0	3	3	3 - 3 = 0	3 = 3 - 0
P <sub>3</sub>	0	8	22	22 - 8 = 14	22 = 22 - 0
P <sub>4</sub>	0	6	14	14 - 6 = 8	14 = 14 - 0

Gantt Chart,



CT(P<sub>2</sub>) CT(P<sub>1</sub>) CT(P<sub>4</sub>) CT(P<sub>3</sub>)

$$\text{Avg waiting time} = (3 + 0 + 14 + 8) / 4 = 6.25 \text{ ms}$$

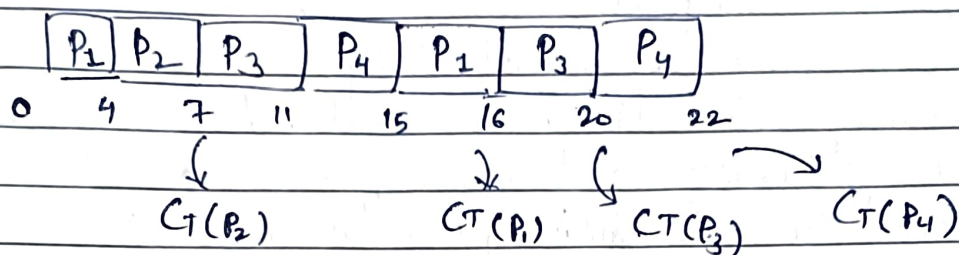
$$\text{Avg turnaround time} = (8 + 3 + 22 + 14) / 4 = 11.75 \text{ ms}$$

(c) Round Robin (quantum = 4ms)

Process	AT	BT	CT	WT	TAT
P <sub>1</sub>	0	5	16	16 - 5 = 11	16 - 0 = 16
P <sub>2</sub>	0	3	7	7 - 3 = 4	7 - 0 = 7
P <sub>3</sub>	0	8	20	20 - 8 = 12	20 - 0 = 20
P <sub>4</sub>	0	6	22	22 - 6 = 16	22 - 0 = 22



Grant Chart,



$$\text{Avg waiting time} = (11 + 4 + 12 + 16) / 4 = 10.75 \text{ ms}$$

$$\text{Avg turnaround} = (16 + 7 + 20 + 22) / 4 = 16.25 \text{ ms}$$

⇒ SJF gives lowest waiting and turnaround times, while RR improves fairness for interactive tasks.

Que 9.

Ans. → (i) Cloud migration:

For a virtualized cloud, a microkernel is suitable, since it separates core services and provides better security and scalability. Virtual machines (VMs) add isolation by running multiple OS instance on the same hardware. They also support resource sharing, load balance and living migration.

(ii) Smart home system:

Here, many IoT devices run together. The OS uses priority scheduling and IPC, so urgent tasks like intrusion detection get

Date.....

CPU immediately, while low-priority scheduling and I/O so urgent tasks like intrusion detection get CPU immediately, which low-priority tasks like lighting wait. Algorithms such as EDF (Earliest Deadline first) or RMS (Rate Monotonic Scheduling) can ensure critical tasks meet their deadlines without starving background process.