# Assignment - 01.

Ans - 1 Almost all the operating system have user

(1) Interface varies b/w command line & graphic user interface batch.

(2) The system is able to load a program into memory and run the program and executed either normally or abnormally.

(3) OS helps to keep track of which user and how much and what kind of computer system.

(4) OS ensure that all access to system resources are in control.

(5) An operating system manages all other application and program in computer.

(6) Application are written for OS API rather than raw hardware making the portable.

Ans 2 For designing a OS for a wearable health device that monitors heart rate. I would prefer using Real Time OS.

Reason:
It is designed to process data and execute task within a guarantee time frame. It is used in system where timing is critical.

Key features.
Predictable response time.
Task prioritisation and scheduling
Minimal latency
Reliability and stability.

Ans. 3 To build a new OS Kernel for a performance critical environment. The structure we would avoid kernel structure.

(1) A microkernel minimize kernel size by moving drivers file system and services to user space.

(2) In microkernel, most services run in user space rather than kernel space.

(3) They prioritize modularity fault isolation and security over raw performance.

(4) They are not best choice when max performance is required.

(5) Each request involves multiple context switches (user ⟷ kernel mode) which add latency.

(6) For performance - critical system this overhead can degrade throughput.

Ans 4. OS structure doesn't just determine whether process run it shapes how they run, how fast, how safely, how easy the system is to change whether it suits latency, security

Ans 5.(i) The PCB stores: process's state, register value program counter and stack pointer during context switch. By analyzing it we can detect errors such as:

(1) Incorrect process state.

(2) wrong PC / SP values.

(ii) Context switching when task moves from running to waiting:

Context switching involves saving the current process state (PC, SP, registers) into its PCB updating its state to waiting the loading the PCB of the next ready process by

restoring it saved state so the CPU can continue execution from where the process left off.

(iii) Allocating I/O resources mid-execution
You should use a blocking synchronous system call because process must wait until the OS allocates the I/O resources safely. Blocking ensures the request is completed before execution resumes - preventing race conditions and resource conflicts.

Ans.6 Save state - 2ms.
Load state - 3ms
Scheduler overhead - 1ms.

(a) Context switching time →
Context switching time = Save state + Load state + Scheduler state.
= 2 + 3 + 1 = 6ms.

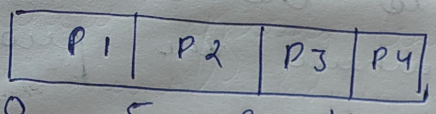
Ans.7. Single Threaded time = 40s
Thread per process N = 2
Execution Time = $\frac{T}{N} = \frac{40}{2} = 20s$.

Speed = $\frac{T_1}{TN} = \frac{40}{20} = 2$

Efficiency = $E = \frac{S}{N} = \frac{2}{2} = 1.0$ (100%).

Ans.8. (a) Gantt Chart
Order : P1 → P2 → P3 → P4.
↳ FCFS (First come first serve).

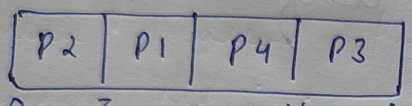
P1 - 0-5
P2 - 5-8
P3 - 8-16
P4 - 16-22




(2) Non-preemptive SJF (Shortest Job First)

Order: P2(3) → P1(5) → P4(6) → P3(8)

- P2 : 0 - 3
- P1 : 3 - 8
- P4 : 8 - 14
- P3 : 14 - 22.



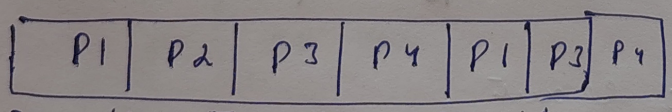

(3) Round Robin (Quantum = 4ms)

Execution order.

→ P1(4) → P2(3) → P3(4) → P4(4) → P1(1) → P3(4)
→ P4(2)

Timeline

- P1 : 0 - 4
- P2 : 4 - 7 (done)
- P3 : 7 - 11
- P4 : 11 - 15
- P1 : 15 - 16 (done)
- P3 : 16 - 20 (done)
- P4 : 20 - 22 (done)




(b) Average waiting Time (WT) & Turnaround time (TAT).

Formulas:

- TAT = Completion Time − Arrival time
- WT = TAT − Burst time.

① FCFS

∴ PN : CT

| Operation | Completion Time | Turnaround Time | Waiting Time |
|---|---|---|---|
| P1 | 5 | 5 | 0 |
| P2 | 8 | 8 | 5 |
| P3 | 16 | 16 | 8 |
| P4 | 22 | 22 | 16 |

Avg TAT = (5 + 8 + 16 + 22) / 4

= 12.75.

(C) Best Algorithm :

→ ~~SJ~~ SJF gives lowest average waiting and turnaround time.

→ Hence it balances throughput and turnaround best among the 3.

9 (i) Cloud migration OS architecture

Architecture :

- Microkernel - because

→ Minimal code increase security

→ Services run in user mode, improving fault isolation.

→ Easy scalability by adding services as modules.

Virtual Machine Now :

→ Provides isolated environment for each service.

→ Enable resource pooling and dynamic allocation.

→ Simplify system management and migration

without affecting others.

(ii) Smart home system scheduling.

Process scheduling + IPC:

→ OS assigns higher priority to critical tasks

→ Lower priority for uncritical tasks.

→ IPC ensures quick communication b/w process.

Suitable Algorithm:

→ Priority scheduling.

→ Earliest Deadline first

→ Round Robin.