usage, efficient power management, and reliability, making it ideal for resource-constrained devices like wearables.

Sol 3) In a performance critical environment, the structure i would avoid is the Microkernel.

While microkernels provide better modularity, and fault isolation, they suffer from performance overhead because most services (like device drivers, file systems, networking) run in user space and communicate with the kernel via inter-process communications (IPC). These frequent context switches and message passing add significant latency, which is unsuitable for performance-critical systems. Instead, Monolithics Kernels or optimized layered Kernels are prefered in such environments since they allow faster systems calls and lower overhead, making them more efficient for performance-demanding tasks.

Sol 4) A developer claims that OS Structure doesn't matter as long as processes run. This claim is not valid because the structure of an OS directly impacts performance, reliability, scalability, and security. A monolithic kernel, for example, may execute processes faster due to fewer communication overheads but risks stability since a single bug can crash the whole systems. On the other hand, microkernels improve modularity and fault isolation but can incure higher overhead during interprocess communication. Thus, while processes may "run" in any structure, the efficiency, maintainability and safety

Operating System Assignment-1

23010102210
Harshit
B.Tech CSE Sec-4

Sol 1) Even with advance hardware, modern systems still rely on Operating Systems because hardware alone cannot provide coordination or ease of use. The OS acts as a bridge between hardware and applications by abstracting and complex operations and offering simple interfaces. It efficiently manages resources like CPU, memory, and I/O devices, while also enabling multitasking and smooth process execution.

In addition, the OS ensures system security and protection through user authentication, permissions and process isolation. It handles device drivers for varied hardware, ensures applications portability across platforms, and provides a user-friendly environment through GUI's and utilities. Thus, the OS remains essential for simplifying hardware interactions, managing resources, and delivering a secure, usable computing experience.

Sol 2) for a wearable health device that monitors heart rate, the most suitable type of operating system would be a Real Time Operating System (RTOS).

An RTOS is designed to process data and respond to inputs within strict time constraints, which is crucial for health monitoring. In a wearable devices, heart rate must be collected, processed and sometime alerted in real time without delays, since late responses could impact the user's health and safety. RTOS provides features like deterministic task scheduling, low memory

Sol6) i) PCB holds registers, program counter, and process state. By checking it, errors like wrong register values or incorrect state transitions (eg: ready vs waiting) can be identified

ii) Context switching saves CPU state of the running process into the PCB, updates its states, picks another process, and restores its saved state into the CPU.

iii) use non-blocking / asynchronous system calls so execution continues while I/O is allocated, avoiding stalls and improving efficiency.

Sol6) a) Total Context Switching Time

Total time = Save State + load State + Scheduler overhead.
= 2 ms + 3 ms + 1 ms = 6 ms

b) Impact on multitasking performance

- Each context switch consumes CPU time without doing useful work.
- Higher switching time → more overhead → less CPU time for processes → reduced performance.
- If too frequent, it slows down multitasking responsiveness.

Sol7) Execution time (ideal multithreading): If work is perfectly divided among threads,

$$\text{Execution time} = \frac{\text{Single-threaded time}}{\text{Number of threads}}$$

$$T = \frac{40}{n} \text{ seconds}$$

How multithreading improves performance:

- Thread run concurrently, utilizing CPU cores better.
- Reduced idle time by overlapping computation and I/O.
- Leads to faster execution and better resource utilization.

Ques 8) Scheduling

Processes : P1 = 5, P2 = 3, P3 = 8, P4 = 6

a) Gantt charts

- FCFS : P1 (0-5) → P2 (5-8) → P3 (8-16) → P4 (16-22)
- SJF : P2 (0-3) → P1 (3-8) → P4 (8-14) → P3 (14-22)
- RR (Q=4) : P1 (0-4) → P2 (4-7) → P3 (7-11) → P4 (11-15) → P1 (15-16) → P3 (16-20) → P4 (20-22) → P3 (20-22)

b) Avg WT & TAT :

- FCFS → WT = 7.25, TAT = 12.75
- SJF → WT = 6.25, TAT = 11.75
- RR → WT = 11.25, TAT = 16.75

c) SJF is best → lowest waiting and turn around.

Ques 9) Virtualization & IOT

i) Cloud migration

- Architecture : Microkernal → secure, modular, scalable.
- VMs help : isolation (fault don't spread), management (easy migration), optimized resource use.

ii) Smart Home IoT

- OS role : uses scheduling + IPC & urgent tasks (intrusion detection) run first, lights etc. handled later.
- Algorithms : Priority scheduling (urgent first), Round Robin (fairness), EDF (real-time deadlines).