

Operating System
Capstone Assignment

Hanshit Chaurasia
2301010 210

Ans1} OS Reliance: Modern systems rely on OS for abstractions that manage complex hardware and simplify programming.

→ Process, Memory, I/O management: OS provides process abstraction, Virtual Memory abstraction, and I/O abstraction (via drivers)

Ans2} OS Structure: Microkernel is best for a distributed web application

→ Justification: High reliability and maintainability due to modular design, where service failures don't crash the minimal kernel.

Ans3} Thread Efficiency: Correct threads are more efficient than processes.

→ Reason: Threads share the processes's address space, context switching is faster and resource usage is lower than creating a new Process Control Block (PCB)

Ans4} Memory Allocation (Processes: 12MB, 18MB, 6MB; Block: 20MB, 10MB, 15MB):

→ first-fit: P₁(20MB), P₃(15MB), P₂(18MB) unallocated. High internal (17MB) and External (19MB) fragmentation

→ Best-fit: P₁(15MB), P₂(20MB), P₃(10MB). All allocated. Low Internal (9MB) fragmentation.

Ans5} Scheduling (P₁: 5ms @ 0, P₂: 3ms @ 1, P₃: 8ms @ 2, P₄: 3ms @ 3):

• Gantt Charts.

→ FCFS = |P₁(5)| P₂(3) | P₄(3) | P₃(8) | Total (19).

→ RR (q=4) : | P₁(4) | P₂(3) | P₃(4) | P₄(3) | P₁(1) | P₃(4) | Total (19)

- Avg Times: SJF has the lowest Average waiting time (0.25 ms) and Turnaround Time (4.25 ms).
- Best Balance: SJF has Best Balance of throughput (low ATT) and efficiency.

n56} Deadlock in Banking:

- a) Banker's Algorithm: Checks if granting a lock request leaves the system in a safe state, preventing the dangerous condition. Requires knowing maximum needs.
- b) Detection and Recovery: Detect deadlocks by checking for cycles in the wait-for graph (WFG). Recover by selecting a victim transaction and performing rollback/termination.

n57} Procedure- Consumers with Semaphores:

- use mutex for mutual exclusion when accessing the buffer capacity for the procedure, full counts items for the consumer.

n8} Page Replacement:

- FIFO: 5 page faults (Replace the oldest page: 2, 1, 4, 2, 3 (replaces 2), 4 (replaces 1), 3)
- LRU: 5 page faults (Replace the least recently used page: 2, 1, 4, 2, 3 (replaces 1), 4 (replaces 2), 3)

n9} Difficulties of concurrent system Design:

Q8) Distributed file Systems:

- a) Critical Issues: Cache Coherence (ensuring all copies of a file are consistent) and Transparency / Naming (providing a unified, location-independent file view).
- b) Architectural Approaches: Client Server Model (eg NFS, using client caching for speed) and clustering / Symmetric Distribution (for high parallelism and fault tolerance).

Ans 10) Synchronous Checkpointing:

→ Mechanism: All processes coordinate and stop to take a local checkpoint simultaneously, creating a single consistent global state for failure recovery

→ Evaluation: Strength → Guaranteed consistency. Weakness - High overhead and low performance due to required coordination

Ans 11) Smart Home Design:

- a) scheduling strategy: Preemptive Priority Scheduling. Justification prioritizes mission-critical task by immediately interrupting low priority tasks to ensure real-time responsiveness.
- b) IPC Methods:
 - Shared Memory: Used for high throughput data transfer (eg raw sensor data) due to speed and minimal system calls.
 - Message Passing: Suitable for distributed / loosely coupled device communication and control commands.

Anst2) Linux Case Study (System Call).

- System call : Using python's os.open() (which map to the Linux open() system call) to create a file.
- Relevance : Demonstrate the OS's System Call Interface, which forced a mode switch from user mode to kernel mode to execute privileged operations (like disk access), ensuring resource protection and controlled hardware access.