

Operating System
Capstone Assignment

Hanshit Chaurasia
2301010210

Ans 1) OS Reliance: Modern system 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)

Ans 2) 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.

Ans 3) Thread Efficiency: Correct Threads are more efficient than processes.

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

Ans 4) Memory Allocation (Processes: 12MB, 18MB, 6MB; Block: 20MB, 10MB, 15MB):

→ first-fit: P1 (20MB), P3 (15MB), P2 (18MB) unallocated. High internal (17MB) and external (14MB) fragmentation

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

Ans 5) 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 Time: STF has the lowest Average Waiting time (1.25 ms) and Turnaround Time (9.25 ms).
• Best Balance: STF: Best Balance of throughput (low ATT) and efficiency.

Ans 6) 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.

Ans 7) Procedure - Consumer with Semaphores:

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

Ans 8) Page Replacement:

→ FIFO: 3 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)

Ans 9) ~~Distributes~~ OS & Concurrent system Design.

Q9) Distributed File System:

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 distributed / symmetric Distribution (for high parallelism and fault tolerance).

Q10) 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: prioritized mission-critical tasks 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.

Ans 12) 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.