Purna Kandpal
23010102851
Btech CSE 'D'

Date ..../..../.....

Assignment - 5

Ques 1 :- The OS provides essential abstractions like processes, virtual memory, and device-independent I/O, hiding hardware complexity.
It ensure resource management, protection, concurrency and stability which applications alone cannot handle.

Ques 2 :- (a) Monolithic → fast but less reliable; one bug can crash system.
(b) Layered → More organized; moderate performance
(c) Microkernel → Highly reliable, modular, easier to maintain

Best for distributed web apps :- Microkernel, because it isolates failures and supports easy updates

Ques 3 :- Threads are more efficient because they share memory and have lighter context switching than processes with large PCBs and separate address spaces.
However, processes offer better isolation, so efficiency depends on the use case.

Ques 4 * First-Fit :-
- 12 MB → 20 MB
- 18 MB → not allocated
- 6 MB → 10 MB

→ fragmentation = 12 MB

* Best-Fit :-
- 12MB → 15 MB
- 18MB → 20 MB
- 6MB → 10MB

→ fragmentation = 9MB

Date ..../..../.....

Ques :- (a) The Banker's algorithm checks whether granting a lock request keeps the system in a safe state. It only allows a transaction to lock more accounts if the remaining resources can still guarantee that every transaction will eventually finish.

(b) Detect & Recover (Programming Approach)
Use a wait-for graph to detect cycles indicating deadlock. On detection, abort or roll back the lowest-priority transaction, release its account locks, and restart it safely.

Ques 9 :- (a) 1. Transparency & Consistency
- Ensuring all users see the same file version across locations is difficult.
- Replication and network delays can cause inconsistent reads/writes.

2. Fault Tolerance & Resource Coordination
- Servers, network, or sites may fail, affecting file access.
- The OS must manage resource location, recovery and load balancing across distributed nodes.

(b) Architectural Approaches for Distributed File Management

1. Client-Server Model
- Servers store files; clients request operations via RPC.
- Caching, replication, and stateless servers improve performance and scalability.

Spiral

Date ..../..../.....

2. Distributed Naming + Replicated File Service
- use global naming for uniform access, and replicate files across sites.
- Replication with version control ensures high availability and reliability even if a site fails.

Ques 11. (a) Preemptive Priority Scheduling
- Give security devices the highest priority.
- Less - critical tasks get lower priority.
- User ISR (top-half) for quick interrupt handling and a high-priority worker thread (bottom-half) for full processing.

Reason :- Ensures security events preempt all other tasks → minimum latency.

(b) 1. Message Queues
- for sending prioritized alerts / events.
- Reason :- Reliable and supports priority ordering.

2. Shared Memory + Mutex
- For fast data sharing (sensor states)
- Reason :- low latency; mutex ensures mutual exclusio

3. Publish-Subscribe (MQTT)
- For communication with cloud / remote devices.
- Reason :- Scalable and light weight for IOT.

4. Signals / Events Flags
- To notify tasks of security interrupts.
- Reason :- Fast one-to-many notifications.

Spiral