Name- Gurleen Kaur
class- Btech CSE (Sec-4)
RollNo- 2301010207

Date
DELTA Pg No.

Capstone Assignment

Operating System

1. Modern system rely on OS because it provides abstractions that simplify hardware use.
   - Process Management: CPU Scheduling, Context Switching, isolation.
   - Memory Management: Virtual Memory, allocation, protection.
   - I/O Management: device drivers, buffering, caching.

2.

Monolithic	Layered	Microkernel
• all services in kernel • fast but less reliable.	organized modules. better maintainability.	minimal kernel + services in user space highly reliable

Best for web app: Microkernel

3. Threads & Processes
   - Threads share memory → light weight.
   - Processes have separate address spaces block smaller.
   - PCB: processes PCB larger; thread control block smaller.
   - Context switching: thread switching faster.
   - Resource Usage: threads needs fewer resources.

4. First-Fit & Best-Fit Allocation.
- Processes: 12MB, 18MB, 6MB
- Blocks: 20MB, 10MB, 15MB.

• First fit
12MB → (20MB) (left 8MB).
18MB → Cannot fit.
6MB → 10MB (left 4MB)

• Best-fit
12MB → 15MB (left 3MB)
18MB → 20MB (left 2MB)
6MB → 10MB (left 4MB)

• Fragmentation: Best fit leaves smaller holes

Part-B

Qu5. Scheduling
- FCFS: long waiting for later process; higher waiting time
- SJF: minimizes average waiting & turnaround.
- RR: fair, good response time, moderate turnaround.

Best Balance & in multiprogramming!
RoundRobin → fairness + responsiveness.

Qu6. Dead locks in Banking.
a) Banker's algorithm:
checks if system is in safe state before granting a lock; allocates resources only if transactions can finish without ~~lock~~ deadlock.

(b) Detection & Recovery:
- Maintain 'wait-for graph.
- Detect cycles.
- Recover by aborting back one transaction.

7. Producer - Consumer
Use:
- mutex for mutual exclusion.
- empty, full semaphores for buffer control.

```
mutex.acquire()
# Critical section
mutex.release()
```

8. FIFO & LRU Page Replacement
Sequence: 2, 1, 4, 2, 3, 4, 3.
- FIFO: replaces oldest page.
- LRU: replaces least recently used Page.

Part-C

9. Distributed file System.

a) Critical issues:
- Consistency: Same file views accross locations.
- Fault tolerance: Server / client failures

b) Architectures
- Client-Server with Caching.

- Distributed namespace

10. Sychronous checkpoint.
- All processes checkpoint at the same time → Consistent ~~gvo~~ global state.
- Recovery ~~is~~ uses last global checkpoint

Strength = Consistent, Simple recovery
Weakness = blocking, high overhead

11. IOT Smart Home Scheduling

a) Use a priority Scheduling.
- highest priority → security devices.
- lower priority → lighting, appliances

b) IPC methods.
- Message queues → asynchronous device communications.
- Shared memory + Semaphores → fast data sharing.
- sockets → remote IOT devices interactions

12. System Call Example (Python)

eg:

```
f = open("example.text", "w")
f.write("Hello OS")
f.close()
```