Date

Name :- Ritu Rajput
Roll No :- 2301010208.

Capstone Assignment

OPERATING SYSTEM

Ans:- Modern system still depend heavily on operating system because an os provides essential abstractions and management services that hardware alone cannot offer.

Process Management.

- Creates, schedules and terminates process.
- Provide CPU scheduling
- Handles context switching.

Memory Management

- Provide virtual memory abstraction
- Allocates memory
- Perform paging, segmentation.

I/O Management.

- Abstract hardware device using device drivers.
- Provides uniform

Date

Ans2. Monolithic kernel

- Entire OS runs in kernel mode
- Fast performance
- Difficult to maintain

Layered OS

- OS divided into layers (hardware → user interface)
- Easier debugging and maintenance.
- Slower due to overhead b/w layer.

Microkernel

- Only minimal functionality in kernel
- Other services run in user mode
- Highly reliable, modular.
- Slight message-passing overhead.

Best choice for distributed web application

- High reliability
- Better maintainability and updates
- Supports distributed service using message.

Ans3. Reasons threads are more efficient.

- Lower Overhead :- Creating & switching threads is faster than processes.
- Shared memory :- Threads share code/data

Date

But there are issues

- Synchronization complexity: Shared memory cause race conditions.

- Security risk :- A faculty thread may corrupt entire process.

- No isolation :- Unlike processes, threads cannot be protected from each others.

Ans:- Process requirement :- 12MB, 18MB, 6MB,
Available blocks : 20MB, 10MB, 15MB.

First fit :-

① 12MB → goes to 20MB Block → leftover = 8MB
② 18MB → not fit in 10 or 15 → alloc. fails.
③ 6MB → goes to 10MB block → leftover = 4B.

Fragmentation :-
- External : 8MB, 4MB, 15MB remain unused.
- Major issue: 18MB cannot be allocated.

Best fit :-

① 12MB → best fit = 15MB block → leftover = 3MB
② 18MB → best fit = 20MB block → leftover = 2MB
③ 6MB → best fit = 10MB block → leftover = 4MB

Date

Fragmentation :-

- Small external fragments : 3MB, 2MB, 4MB

Ans Given :-

Process	Burst	Arrival
P1	5	0
P2	3	1
P3	8	2
P4	6	3

(a) Gantt charts

FCFS

P1	P2	P3	P4
0	5	8	16	22

SJF

P1	P2	P4	P3
0	5	8	14	22

Round Robin (q = 4

P1	P2	P3	P4	P1	P3	P4	P3
0	4	7	11	15	16	18	22	24

Date

(b) Waiting Time and Turnaround Time

FCFS

- WT = [0, 4, 6, 13] → Avg WT = 5.75
- TAT = [5, 7, 14, 19] → Avg TAT = 11.25

SJF

- WT = [0, 4, 5, 8] → Avg WT = 4.25
- TAT = [5, 7, 11, 14] → Avg TAT = 9.25

RR (q=4)

- WT = [11, 6, 14, 9] = Avg WT = 10
- TAT = [16, 9, 22, 15] = Avg TAT = 15.5

Ans 6.

(a) Banker's Algorithm:-

- checks Available resources before allocation.
- Ensures System always stays in a 'safe state'.

(b) Detect & Recover Approach.

- Use wait for graph to detect cycles
- on deadlock → abort lowest priority transaction or roll back.

Date

Ans 7. Producer - consumer using Semaphores (Python)

```
import threading, time
import random

empty = threading.Semaphore(5)
full = threading.Semaphore(0)
mutex = threading.Semaphore(1)
buffer = []

def producer():
    while True:
        item = random.randint(1, 100)
        empty.acquire()
        mutex.acquire()
        buffer.append(item)
        mutex.release()
        full.release()

def consumer():
    while True:
        full.acquire()
        mutex.acquire()
        item = buffer.pop(0)
        mutex.release()
        empty.release()
```

Date

Ans 8. FIFO & LRU Page Replacement

Sequence: 2, 1, 4, 2, 3, 4, 3

- FIFO: replaces oldest page
- LRU: replaces least recently used Page.

Ans 9. Distributed file system

(a) Critical issues
- Consistency: same file view across locations.
- Fault tolerance: server / client failures.

(b) Architectures.
- client – server with caching.
- Distributed namespace.

Ans 10. Sychronus checkpoints
- All process checkpoint at the same time → consistent global state.
- Recovery uses last global checkpoint strength = consistent, simple recovery.

Ans 12. System call Example (Python)

Eg :-

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