Ans4) O/S Hardware Interaction

→ Hardware (MMU) : Performs the real-time translation using the PTBR · set by the OS.

→ TLB : A fast cache for recent translations, crucial for speed.

→ Protection : Achieved via Valid / Invalid bits and Protection bits within the Page Table Entries.

Ans5) Calculations (16 bit VA, 1KB Page)

→ a) Number of Virtual Pages :

$$\text{Pages} = \frac{2^{16}\text{ bytes}}{2^{10}\text{ bytes}} = 2^6 = 64$$

→ b) Page Table Size (2-byte entry) :

$$\text{size} = 64 \text{ entries} \times 2 \text{ bytes/entry} = 128 \text{ bytes.}$$

Ans6) Memory Allocation

a) & b) Step- by Step Allocation and Unused Memory

1 First -fit Allocation (FF)

Step	Process	Size (KB)	Allocation Action	Free Memory ~~Allocation~~ Holes
1	P1	212	Allocated P1 from 1000 KB	788
2	P2	417	Allocated P2 from 788KB	371
3	P3	112	Allocated P3 from 371KB	259
4	P4	426	Cannot Allocate 426 > 259	259

Operating System
Assignment 2

Harshit Chaurasia
2301010210

Ans1) Address Translation (Paging)

→ Mechanism : Converts logical Address <p, d> (CPU output) to physical address <f, d> (RAM location) using the page table.

→ purpose : Memory protection and enabling non- contiguous allocation.

Ans2) Fragmentation

→ Internal : Wasted space within an allocated block (eg- unused space in the last page)

→ External : Free space is scattered in small non-contiguous chunks.

→ Mitigation
  · Paging : Eliminates external, minimises internal.
  · Buddy System : Merges adjacent free blocks to reduce external fragmentation.

Ans3) Paging Model Trade - offs.

→ Model : Hierarchical Paging to manage large adress spaces.

→ Overhead : High (requires multiple page tables per processes)

→ Speed : Slower but mitigated by the TLB

→ Fragmentation : Low External ; Low Internal

Optimal	6	Best Performance, as it removes the page needed furthest in the future.
LRU	7	Performs better than FIFO in general, but equal here. Does not suffer from Belady's Anomaly.

Ans 8) Demand Paging Overhead

a) <u>Additional Time Overhead</u>

→ Dirty Pages = 1000 × 30% = 300 pages

→ Overhead = 300 × 10ms / Page = 3,000 milliseconds.

b) <u>Optimization Technique</u>

→ Prepaging : Use a background Process to pre-clean dirty pages by writing them to disk asynchronously. This eliminates the mandatory 10 ms synchronous disk write delay during a page fault.

Ans 9) <u>Autonomous Vehicle Case Study</u>

a) Working Set Model & Page Replacement.

→ Working Set Model : used to determine the minimum required pages for each task.

→ Strategy : Mission-Critical is guaranteed its full working set size to prevent thrashing and ensure low latency. Less critical pages are prioritized for replacement when system memory is tight.

b) <u>Memory Allocation Strategy</u>

→ Strategy : Fixed Allocation with Global Page Replacement (or Priority-Based Allocation).

26/11/25

∴ Total Allocated: 212 + 417 + 112 = 741 KB

b) Total Unused Memory (FF) : 259KB

b),2) Best - Fit Allocation (BF) :

Step	Process	Size (KB)	Allocate Action	Free Memory Holes
1	P1	212	P1 from 1000KB	788
2	P2	417	P2 from 788KB	371
3	P3	112	P3 from 371KB	259
4	P4	426	Cannot Allocate 426 > 259	259

∴ Total Allocated = 741KB

∴ b) Total Unused Memory : 259KB

c) Identify which method gives the best utilization and explain why.

Method	Total Allocated Memory (KB)	Total Unused Memory (KB)
First-fit	741	259
Best-fit	741	259
Worst-fit	741	259

∴ All three methods provide the same utilization

Ans 7) Page Replacement Simulation & Analysis

Algorithm	Page fault (3 frames)	Comment
FIFO	7	Susceptible to Belady's Anomaly