OS

Answer 1

Address translation is the process of converting a logical (virtual) address generated by the CPU into a physical address in main memory.
Each process has its own page table managed by the OS.

Steps

1) CPU generates a logical address → (Page number + offset)

2) The MMU (Memory Management Unit) uses the Page table to find the corresponding frame number.

3) Combine Frame Number + offset → Physical Address.

CPU → Logical Address → [MMU + Page Table] → Physical Address → Main Memory.

Answer 2

Internal Fragmentation: Occurs when memory is allocated in fixed-sized blocks, and some space inside the block remains unused.

External Fragmentation: Happens when free memory is scattered in small non-contiguous blocks.

Example layout

| P1 (100KB) | FREE (10KB) | P2 (200KB) | P3 (300KB) | FREE (15KB) |

- Internal fragmentation = space wasted inside allocated blocks.
- External fragmentation = total free space is enough, but not contiguous

answers: In paging, memory is divided into fixed-size blocks

- Pages (logical memory) and Frames (physical memory)

Model Example

- Page size = 1KB
- Logical memory = 16KB → 16 pages
- Physical memory = 8KB → 8 frames

Trade-offs

Aspect	Advantage	Disadvantage
Fragmentation	no external fragmentation	Internal fragmentation may occur
Memory access	Simplifies allocation	slower due to page table lookup
Overhead	Easy swapping	page table consumes extra memory

Answer: Virtual memory allows execution of processes larger than physical memory by storing parts on disk.

OS Role
Maintains page tables
Handles page faults and swapping.

Hardware Role
MMU (Memory Management Unit) perform address translation
TLB (Translation lookaside Buffer) caches recent transactions for speed.

TLB (Translation lookaside Buffer) caches recent translations for speed.

Answer: a) Determine no. of virtual pages

Virtual address space = 2^{16} = 65,536 bytes = 64KB
Pagesize = 1KB = 2^{10} bytes.

a) No. of pages = $\frac{64KB}{1KB}$ = 64 Pages

no. of virtual pages = 64

b) Pagesize = 64 entries x 2 bytes = 128 bytes

Answer 6 Total memory = 1000 KB

Process	Size (KB)
P1	212
P2	417
P3	112
P4	426

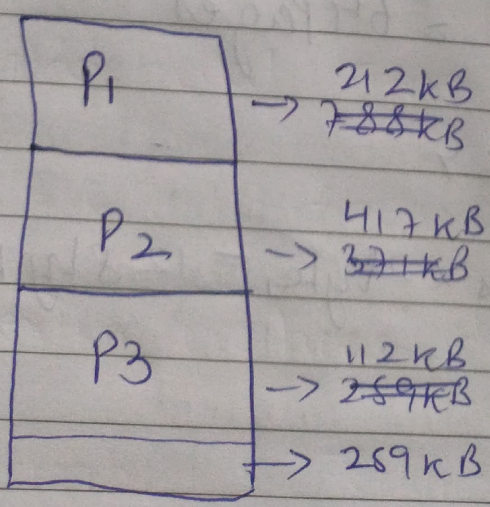
Assume First Fit Algo.

① Allocate P1 = 212 KB → Remaining = 1000 − 212 = 788 KB

② Allocate P2 = 417 KB → Remaining = 788 − 417 = 371 KB

③ Allocate P3 = 112 KB → Remaining = 371 − 112 = 259 KB

④ Try to allocate P4 = 426 KB → Not enough. (259 KB left) → P4 cannot be allocated.



c) Best utilization

All three algorithms give same utilization in this case because memory blocks are contiguous. Best fit is generally preferred when minimizing unused space in real systems.

Answer 7

FIFO

	7	0	1	2	0	3	0	4	2	3	0	3	2
		0	0	0	0	3	3	3	2	2	2	2	2
7	7	7	2	2	2	2	4	4	4	0	0	0	
*	*	*	*	HIT	*	*	*	*	*	*	HIT	FIT	

7 0 1 2 0 3 0 4 2 3 0 3 2

Page Fault - ~~10~~ 9 10

Page HIT - 3

OPTIMAL

	7	0	1	2	0	3	0	4	2	3	0	3	2
		0	0	0	0	0	0	4	4	4	0	0	0
7	7	7	2	2	2	2	2	2	2	2	2	2	2
*	*	*	*	HIT	*	HIT	*	HIT	HIT	*	HIT	HIT	

7 0 1 2 0 3 0 4 2 3 0 3 2

Page fault - 7

Page HIT - 6

d LRU

	1	1	1	3	3	3	2	2	2	2	2
	0	0	0	0	0	0	0	3	3	3	3
7	7	2	2	2	2	4	4	4	0	0	0
*	*	*	HIT	*	HIT	*	*	*	*	HIT	HIT

7 0 1 2 0 3 0 4 2 3 0 3 2

Page Fault - 8
PageHIT - 4

Answers Demand Paging Overhead

Disk write time = 10ms
Memory write time = 100ns = 0.0001ms
30% replaced pages are dirty
1000 pages replaced

a) Tal Additional Time

= 300 x 10 = 3000 ms = 3seconds

3sec extra overhead due to dirty pages

b) Optimization Technique

Use "copy-on-write" and "write-back caching" to delay disk writes

Maintain a dirty bit mechanism to track modified pages efficiently.

- Use background write ~~demon~~ daemon to asynchronously flush dirty pages to disk.

Ans 9 a) Working Set Model Explanation

OS Maintains the working set - the set of pages actively used by each task.

Mission - critical tasks (obj. detection, route planning) get higher priority and are kept resident in memory.

- Less critical tasks (infotainment) use page replacement when memory pressure increases.

- OS uses priority based LRU or WS clock policy to prevent thrashing for critical processes.

b) Memory Allocation Strategy
Use a Hybrid Dynamic Partitioning strategy with:
i) Fixed reserved memory for real time modules (obj