Date .13th.... September 2025

Name - Raghav Kumar Jha
Roll No - 2301010260
Course - B.Tech CSE-D
Subject - Operating System.

Assignment - 2

Q-1 Demonstrate the process of address translation within a modern computing system where multiple process Coexist. Use illustration to clarify the translation from logical to physical memory spaces.

Sol. logical (virtual) addresses are translated into physical addresses using MMU (Memory Management Unit).

- logical Address → divided into Page Number + offset.
- Page Number → mapped using Page Table to Frame Number.
- Physical Address = frame Number + offset.

Q-2 Sol Internal & External Fragmentation.

- Internal Fragmentation : A 100 KB partition used by a 90 KB process → 10 KB wasted.

- External Fragmentation : free memory exists but in scattered blocks

Techniques :
- Paging
- Segmentation
- Buddy System allocation.

Spiral

Date ....

Q-3 Sol

- Memory divided into fixed - sized pages.
- Processes allocated non - contiguous frames.

Trade offs : • overhead - Pages table consumes memory.
• Speed - Address translation slower (solved by TLB).
• Fragmentation : Eliminates external but causes internal fragmentation within last page.

Q-4 Sol OS - Hardware Interaction (Virtual Memory).

- Hardware Support :
  → Page Table → stores mappings.
  → TLB (Translation Looking Buffer) - Speeds up translation.
  → MMU → performs translation

ex - Accessing page not in RAM → OS triggers page fault, loads page from disk.

Q-5 Sol

- Virtual Address = 16 bits → Address space = 2^{16} = 65,536 bytes.

Page size = 1KB = 1024 bytes. = 2^{10}.

Virtual Pages = $2^{16} / 2^{10} = 2^{6}$ = 64 pages.

Each ~~Byte~~ entry = 2 bytes

Page Table Size = 64×2 = 128 bytes.

Date ....

Q-6 $P_1 = 212$ KB $P_2 = 417$ KB $P_3 = 112$ KB

$P_4 = 426$ KB

Step	Action / Algo Rule	Allocated Block (s)	Remaining free block
0	Start	-	1000
1	Allocate $P_1 = 212$	$P_1 \rightarrow 212$	$1000 - 212 = 788$
2	Allocate $P_2 = 417$	$P_1 \rightarrow 212$ $P_2 \rightarrow 417$	$788 - 417 = 371$
3	Allocate $P_3 = 112$	$P_1 - 212$ $P_2 - 417$ $P_3 - 112$	$371 - 112 = 259$
4	Try Allocate $P_4 = 426$	P_4 cannot fit	free 259

Total allocate $= 212 + 417 + 112 = 741$

unused $= 259$

Q-7 (A) FIFO

Ref.	Frames	Page fault	Evicted
7	7, -, -	✓	-
0	7, 0, -	✓	-
1	7, 0, 1	✓	-
2	2, 0, 1	✓	7
0	2, 0, 1	x	-
3	2, 3, 1	✓	0
0	2, 3, 0	✓	1
4	4, 3, 0	✓	2
2	4, 2, 0	✓	3
3	4, 2, 3	✓	0
0	0, 2, 3	✓	4
3	0, 2, 3	x	-
2	0, 2, 3	x	-

Total FIFO Page fault = 10.

Date ....

(B) optimal (Belady's optimal)

Ref	Frames (F1, F2, F3)	Page fault	evicted
7	7, -, -	✓	-
0	7, 0, -	✓	-
1	7, 0, 1	✓	-
2	2, 0, 1	✓	7
0	2, 0, 1	✗	-
3	2, 0, 3	✓	1
0	2, 0, 3	✗	-
4	2, 0, 4	✓	3
2	2, 0, 4	✗	-
3	2, 0, 3	✓	4
0	2, 0, 3	✗	-
3	2, 0, 3	✗	-
2	2, 0, 3	✗	-

Total optimal Page fault = 7

(C) LRU

Ref	frames (F1, F2, F3)	Page fault	Evicted
7	7, -, -	✓	-
0	7, 0, -	✓	-
1	7, 0, 1	✓	-
2	2, 0, 1	✓	7
0	2, 0, 1	✗	-
3	2, 0, 3	✓	1
0	2, 0, 3	✗	-
4	4, 0, 3	✓	2
2	4, 2, 3.	✓	0

Date ....

3	4, 2, 3	x	–
0	0, 2, 3	✓	4
3	0, 2, 3	x	–
2	0, 2, 3	x	–

Total LRU Page faults = 9.

→ Which Perform Best.
- optimal perform best uses perfect future knowledge.
- LRU is practical policy that approximately optimal.
- FIFO offer performs worst and is susceptible to Belady's Anomaly.

Q-8 Sol Disk write = 10 ms, 10,000,000 ns
memory write = 100 ns   Extra time dirty page = 10,000,000 − 100 = 9,999,900
30% of 1000 pages = 300
Total overhead = 300 × 9,999,900 = 3 sec.
Optimization = use write-back with dirty bit or page buffering.

Q-9 (a) Working set model + Policy:
- OS Keeps object detection pages always in memory.
- Infotainment uses page replacements (LRU/Clock) – adapt memory pressure.

(b) Memory Allocation Strategy:
- Priority based dynamic allocation.
- Real time get guaranteed frames.
- Remaining frames used by background tasks.

→ Balances responsiveness with efficiency.

Spiral