Topic .... Date ..../..../.......

PAGE NO.

OPERATING SYSTEMS ASSIGNMENT - 2

Q1. 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

logical (virtual) addresses are ~~+~~ translated into physical addresses using MMU (Memory Management Unit)
- logical Address → divided into page no. + offset
- page number → mapped using page table to frame no.
- physical address = frame no. + offset.

Q2. Internal and External Fragmentation.
- Internal Fragmentation : A 100KB ~~par~~ partition used by a 90 KB process → 10KB wasted

- External Fragmentation : Free memory exists but in scattered blocks

Techniques: → paging
→ segmentation
→ buddy system allocation

Q3.
- Memory divided into fixed - sized pages.
- Processes allocated non- contiguous frames.

Trade offs : • overhead - pages table consumes memory
- speed - address translation slower (solved by TCB).
- fragmentation : eliminates external but causes internal fragmentation within last page.

Teacher's Signature..............................

Topic .... Date ...../...../........

PAGE NO.

Q4 OS - hardware interaction (virtual memory)

- hardware support:

→ page table → stores mappings

→ TLB (Translation Looking Buffer) - speeds up translation

→ MMU - performs translation

eg accessing page not in RAM → OS triggers page fault, loads page from disk.

Q5 • virtual address = 16 bits → address space =

$2^{16} = 65,536$ bytes

page size = 1 KB = 1024 bytes = 2^{10}

virtual pages = $2^{16} / 2^{10} = 2^6 = 64$ pages

each entry = 2 bytes

page table size = $64 \times 2 = 128$ bytes

Q6 $P_1 = 212$ 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 $P_1 = 212$	$P_1 \to 212$	$1000 - 212 = 788$
2	Allocate $P_2 = 417$	$P_1 \to 212$ $P_2 \to 417$	$788 - 417 = 371$
3	Allocate $P_3 = 112$	$P_1 - 212$ $P_2 \to 417$ $P_3 \to 112$	$371 - 112 = 259$
4	Try Allocate $P_4 = 426$	P_4 can't fit	free 259.

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

evicted ~~wasted~~ = 259.

Topic .... Date ..../..../........ PAGE NO.

(B) Optimal ~~Belad~~ (Belady's Optimal)

Ref	Frames (F_1, F_2, F_3)	Page Fault	Evicted.
7	7, -, -	✓	-
0	7, 0, -	✓	-
1	7, 0, 1	✓	-
2	2, 0, 1	✓	7
0	2, 0, 1	X	-
3	2, 0, 3	✓	1
0	2, 0, 3	X	-
4	2, 0, 4	✓	3
2	2, 0, 4	X	-
3	2, 0, 3	✓	4
0	2, 0, 3	X	-
3	2, 0, 3	X	-
2	2, 0, 3	X	-

Total Optimal Page Fault = 7

P.T.O.

(C) LRU

Ref	Frames (F_1, F_2, F_3)	Page Fault	Evicted.
7	7, -, -	✓	-
0	7, 0, -	✓	-
1	7, 0, 1	✓	-
2	2, 0, 1	✓	7
0	2, 0, 1	X	-
3	2, 0, 3	✓	1
0	2, 0, 3	X	-
4	4, 0, 3	✓	2
2	4, 2, 3	✓	0
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 ~~project~~ perfect future knowledge
- LRU is practical policy that approximately optimal.
- FIFO offers platforms worst and is ~~suspects~~ susceptible to Belady's Anomaly.

Topic .... Date ...../...../........

PAGE NO.

Q8 Disk write = 10 ms, 10,000,000 ~~ms~~ ns

memory write = 100 ns

extra time dirty pages = 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.

Q9 a) Working set model + policy:

- OS keeps object detection pages always in memory
- Infotainment uses page replacements (LRU/ clock) - adapt memory pressure