Date ___ / ___ / ______

Name:- Anushka Budia
Roll No.:- 2301010263
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 ~~to~~ + offset.

Q-2 Sol Internal & External Fragmentation:

- ~~to~~ Internal Fragmentation: A 100KB partition used by a 90KB process → 10KB wasted.
- External Fragmentation: Free memory exists but in scattered blocks.

Techniques:
- Paging
- Segmentation
- Building system allocation

Date ___/___/_____

Q-3 sol
- Memory divided into fixed- sized pages.
- Process allocated non- contiguous frames.

Trade offs : • overhead – Pages table consumes memory.
- speed – Address translation slower (solved by TCB).
- Fragmentat : Eliminates external but causes internal ~~fugr~~ 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

Q5 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 entry = 2 bytes
Pages Table Size = $64 \times 2 = 128$ bytes

Date ___/___/___

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 $P_3 = 112$	$P_1 - 212$ $P_2 - 417$ $P_3 - 112$	$371 - 112 = 259$
4	Try Allocate $P_4 = 420$	P_4 cannot fit	free 259

Total allocate $= 212 + 417 + 112 = 741$
unused $= 259$

Q7 (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	×	—
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	×	—
2	0, 2, 3	×	—

Total FIFO Page fault = 10

Date __/__/____

(B) Optimal (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	×	-
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 (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	×	-
3	2, 0, 3	✓	1
0	2, 0, 3	×	-
4	4, 0, 3	✓	2
2	4, 2, 3	✓	0
3	4, 2, 3	×	-
0	0, 2, 3	✓	4

Date ___/___/______

3	0,2,3	x	–
2	0,2,3	x	

Total LRU Page faults = 9

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

Q8 sol Disk write = 10 ms, 10,000 ms
memory write = 100 ms
Extra time disks 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 on pages buffering.

Q9 (a) Working set model & Policy:
- OS keeps object detection pages always in memory.
- Infotainment uses page replacements (LRU/clock) - adopt memory pressure

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

→ Balance responsiveness with efficiency.