Prerna Kandpal
230101025
BTech CSE 'D'

Date ..../..../........

Assignment 02

Ques1 In modern system with multiprogramming, multiple process co-exist in memory. Each process is given a logical (virtual) address space, while the actual data resides in physical memory (RAM). The translation is managed by memory management unit (MMU) with help of page Tables.

i) Logical address generation.
ii) MMU translation via page table.
iii) Physical Address Formation.
iv) Access to physical memory.

Ques2 Memory layout with fragmentation :-
- Internal → Fixed partitions waste space inside blocks.
- External → Free memory is scattered.

* Modern OS solutions (Beyond compaction)
- Paging → removes external fragmentation.
- Segmentation paging → reduce both types.
- Buddy system → split/merges memory in powers of two.

Ques3 Paging-Based memory allocation model :-
- Memory is divided into fixed size pages.
- Each process has a page table mapping virtual pages.
- MMU + TLB handle fast translation : on TLB.
- eliminates external fragmentation (any free frame can be used).

Spiral

Date ..../..../........

* Trade-offs :-
- memory overhead :- Large page tables per process.
- Speed :- TLB hits are fast, misses /page fault slow execution.
- NO external fragmentation : but small internal frag.

Ques4 When an OS manages a virtual memory, it works closely with the hardware to make sure processes run in isolated, protected and efficient environment.

i) Virtual to physical Address translation
ii) Page tables
iii) Translation lookaside Buffer (TLB)
iv) Memory protection
v) Page faults and Demand paging.

Ques5 Virtual address size = 16 bits
page size = 210 → 10 bits used for offset within page
remaining Bits = 16-10 = 6 Bits

so, page no. filled = 6 bits
page offset filled = 10 bits

no. of virtual pages = 2^6 = 64 virtual pages.

(b) Each virtual page needs one entry in page table
no. of entries = 64
each entry = 2 Bytes
page total size = 64×2 = 128 Bytes.

Spiral

Date ..../..../........

Ques 6 Memory Allocation Simulation
A system has 1000 KB of free memory.

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

using first-fit, Best fit and worst-fit

start = 1000 KB
place P1 (212 KB) → remaining = 1000 - 212 = 788 KB
place P2 (417KB) → remaining = 788 - 417 = 371 KB
Place P3 (112 KB) → 371 - 112 = 259 KB
Try, P4 (426 KB) → 259 KB < 426 KB => can't place

Result :- Allocated P1, P2, P3, P3 ; free = 259 KB

Conclusion :- All time produces identical results
their allocation : 259 KB unused
P4 not placed

Ques 7 (a) disk pgs = 0.30 X 1000 = 300
disk time = 300 X 10 ms = 300 ms = 3 seconds
memory time = 1000 X 100ms = 100,000 ms = 0.1 ms

Total time spent = disk time + mem
so additional overhead = 3 sec
total = 300 0 ms + 0.1
= 3000 - 1 ms
= 3 sec

Date ..../..../........

(b) Proposed optimization to reduce this overhead:-

Best single practical optimization
Background pre-clearing - prefer clean victims
Two linked ideas that are commonly used together.

- page-cleaner deamor (Background)
- clean first victim selection

Ques9 Autonomous vehicle case study :-

(a) working set model + replacement policy.
- OS tracks recent active pages per tasks.
- for object detection :- Allows flexible replacement as it adapts to available memory.

(b) Memory Allocation strategy :-
- use - priority - based dynamic allocation
- Real-time responsiveness ensured by working set + real time schedule.

Satinder
21/11/24