SNEHAL KUMAR
2019101003

OS~~&~~NW
QUIZ-6

Q.1. What is the cause of thrashing? How does the system detect thrasing? Once it detects.

A: • Thrashing occurs ~~due~~ when there ~~are~~ is less allocation of minimum pages required by a process. This causes continuous page faults and this ~~cau~~ reduces CPU utilization. ~~Thrashing is also caused~~ (when virtual memory resources are overused.)

• The system detects thrashing by monitoring CPU utilization level compared to the ~~lo~~ multiprogramming level. Thrashing causes reduce in CPU utilization.

• It can be reduced by increasing number of pages alloted or by temporarily suspending the process until sufficient pages are free. ~~Also~~ or by decreasing the level of multi-programming through Working Set Model or Page Fault Frequency ~~These~~ They are based on locality model and prevent thrashing.

Q.2 Discuss the positive and negative aspects of inverted page table approach.

ADVANTAGES.

A: • Inverted page table can be implemented in system with 64 bit architecture easily. Multi~~page~~ level page requires a lot of space and levels in 64 bit. Inverted page reduces memory needed to store.

• Inverted page table stores one entry for each real page of memory ~~addresses~~ instead of all pages stored in multi level

DISADVANTAGES.

• Inverted page ~~stores~~ uses lesser memory ~~and hence is more efficient to search~~ but the time required to search for the page is increased and hence suffers from performance compared to multi-level paging where it is indexed.

• Because of only one entry per frame, we can no longer map the same physical as frame to multiple ~~page~~ virtual page numbers in different processes (shared memory). Multilevel paging doesn't suffer from this as only reference is needed to be passed to multiple processes.

Scanned with CamScanner

Q.3. Assume a page size of 4K bytes and that a page table entry takes 4bytes.

A: Page table ~~size~~ entry size = 4bytes

Page size = 4K bytes
Each page table has 2^{10} ~~entries~~ pages
Number of pages = $4 \times 1024 = 2^2 \times 2^{10} = 2^{12}$

Total addresses = $2^{10} \times 2^{12} = 2^{22}$ bytes (No of pages x page size)

But Address space size = 2^{64} bytes. By adding a second layer of page tables, the top page table would point to 2^{10} page tables, addressing a total of 2^{32} bytes.

on repeating this process, we get, (find minimum required to satisfy 2^{64})

Level 1 → 2^{22} byts
2 → 2^{32} "
3 → 2^{42} "
4 → 2^{52} "
5 → 2^{62} "
6 → 2^{72} bytes.

Thus, 6 levels of page table would be required to map a 64 bit address space.

Q.4. Identify whether the following statements is TRUE or FALSE. If the statement is FALSE correct it and justify the corrected.

A: The statement is TRUE.
Local replacement allows a process to select a victim process from only its own set of allocated frames.
Global replacement allows a process to select a frame from another process and thus if another process has available free frames, it can use it unlike local ~~so~~ allocation. ~~Also~~ Thus, increasing CPU throughput and utilization.
Also the high priority process can take frames of low priority process.
Thus, overall, global allocation algorithm gives letter throughput over local allocation algorithm.

Scanned with CamScanner