Page offset

Session 8

Q.1) What is virtual memory? [swap area]

→ 1) V.M is a memory management technique that provides an "illusion" of a large, contiguous block of memory.
2) It allows an O.S to run large applications or multiple applications simultaneously by using both RAM & disk space.

= • Working:

+ 1) Logical Address space -

→ • Each process is given a virtual address space that is independent of the physical memory available.
• This space can be much larger than the actual RAM, enabling processes to assume they have more memory than it physically exists.

+ 2) Paging:

→ A] Demand paging:
• Virtual memory often uses a technique called "Demand paging", where
[only the necessary parts of pages are loaded into physical memory.]
[Pages not currently in use are stored in disk (swap file)]

B] Page Table:
• A page table maps virtual addresses to physical addresses.
• When a program accesses data that isn't in RAM, a (page fault) occurs, the required page is loaded from disk into a free frame in physical memory.

Scanned with CamScanner

* 3) Swap space:
→ a) V.M uses a portion of hard disk as an extension of RAM.
b) The disk space, often referred to as "swap space" stores pages that are not actively used in RAM.
page swapping → c) When the RAM is full, and a new page needs to be loaded, the OS swaps out an inactive page into the swap space, freeing up RAM for new page.

* 4) Translation lookaside Buffer (TLB)
→ a) To speed up address translation process, modern CPU use (TLB).
b) TLB is small cache that stores the recent translations of virtual addresses to phy addresses.
c) If virtual address is found in TLB, the corresponding physical address is retrieved quickly w/o consulting page table

* 5) Page Replacement Algo
→ a) FIFO (First In First Out)
- Pages are removed in the order they were loaded, regardless of usage.

b) LRU (Least Recently used)
- Pages that have not been used for longest time are replaced first.

c) Optimal algo: Replaces the page that will not be used for the longest time in the future
* It requires prediction & is mainly theoretical

d) MRU (Most recently used
- Page will be replaced which has been used recently (Belady anomaly can occur)

6) ADV:-
→ a) Efficient Memory use:
- Allows the execution of large applications and the simultaneous running of many processes even if physical memory is limited.

b) Isolation:
- Each process is isolated from others as each has its own V.M space
[security, stability ↑]

c) Flexibility:
- Processes can be allocated more memory dynamically.

7) Dis - ADV:
→ A] Performance overhead: (Swap space)
- Accessing data from disk is much slower than accessing RAM, leading to potential performance degradation
- Especially if system is frequently swapping pages in & out (Thrashing)

* Thrashing:
What = When a system spends more time swapping space in & out of memory than executing actual processes.

Scanned with CamScanner

Thrashing

⇒ This happens when there isn't enough physical memory to support the active set of processes, leading to constant page faults & significant slowdown.

* Practical Implementation in Modern System.

1) Windows:
⇒ Uses a page file for swapping
⇒ The size of the page file is often set auto by the system but can be manually configured.

2) Linux/Unix:
⇒ Uses a separate swap partition or swap file
⇒ 'swapon' & 'swapoff' commands manage the use of swap space.

3) MacOS:
⇒ Handles V.M auto & uses swap files within the file system.

* Session 9

(Q.1) Necessary conditions of deadlock.

→ • Deadlock occurs when four conditions are met simultaneously:
a) Mutual exclusion
b) Hold & wait
c) No preemption
d) Circular wait

Scanned with CamScanner

Notes

Q.2) Deadlock prevention & avoidance

→ 1) It involves ensuring that at least one of the four conditions cannot hold & deadlock avoidance, which requires making resource allocation decisions dynamically to ensure a system never enters a deadlocked state.

Notes

Q.3) Semaphore

→ 1) A semaphore is a synchronization tool used to control access to a common resource in a concurrent system.

2) It is an integer variable that can be used to solve critical section problems & to ensure mutual exclusion.

Notes

Q.4) MUTEX:

→ • A MUTEX is a more restricted type of semaphore.

• It is used to lock a resource so that only one thread or process can access it at a time, ensuring mutual exclusion.

Q.5) Producer - Consumer Problem

→ 1) It is a classic sync problem where two processes, the producer & the consumer, share a common buffer.

2) The producer adds items to the buffer & the consumer removes them.

3) Semaphores or mutexes are typically used to manage access to the buffer & ensure that the producer doesn't overflow the buffer & the consumer doesn't try to consume an item when buffer is empty.

Scanned with CamScanner