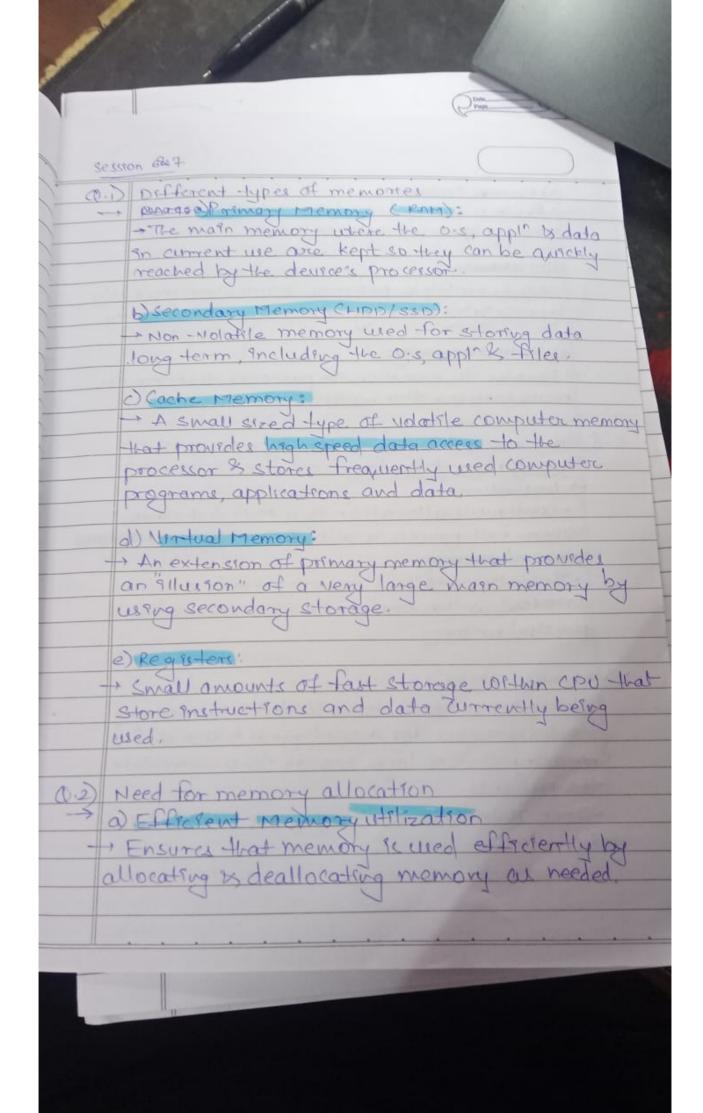
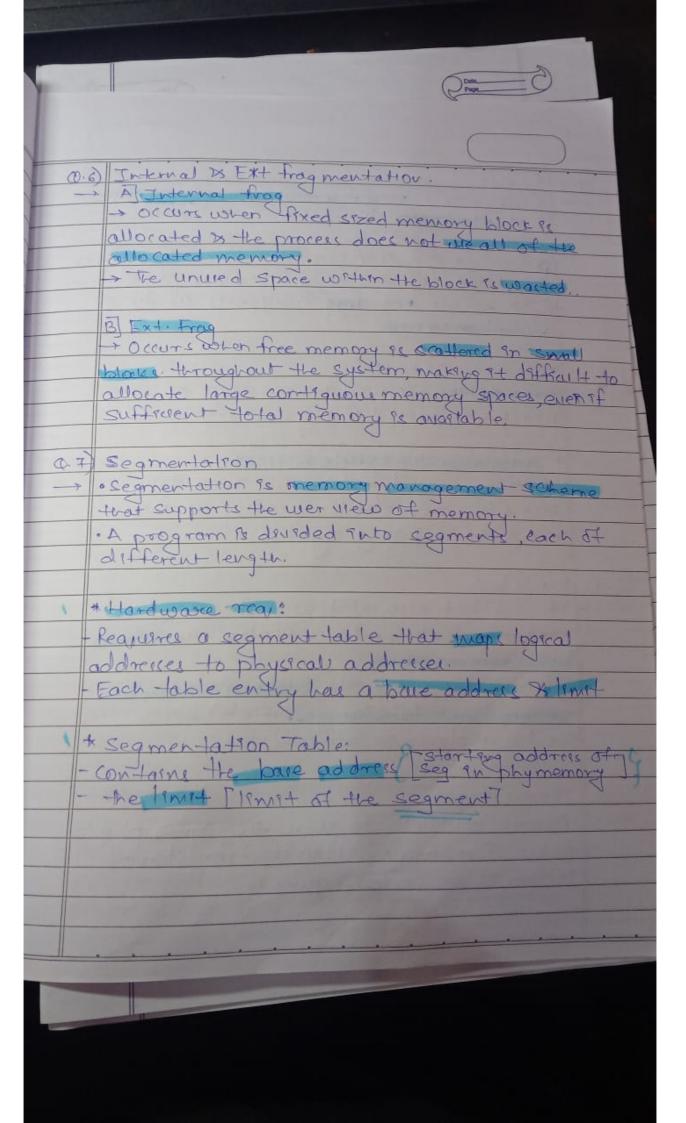
-> Protects each process from interfering with the memory space of another process Allows nultiple processes to run semultaneously by managing memory allocation & de allocation - Frables the execution of processes that require more memory than what se phyercally available. (0.3) Contiquous & Dynamic Allocation in M.M. -> A configuous Allocation. · contiguous allocation es a M.19 technique -> Definition: where each process 9s allocated a single contiquous block of memory. . The entire process's memory needs are catisfred within one continous section of physical memory. -> How 1+ works?: · When a process is to be loaded into memory the O.s searches for a configuous block of free memory that is large enough to accompodate the process. · Once found, the entire process is loaded into this block.

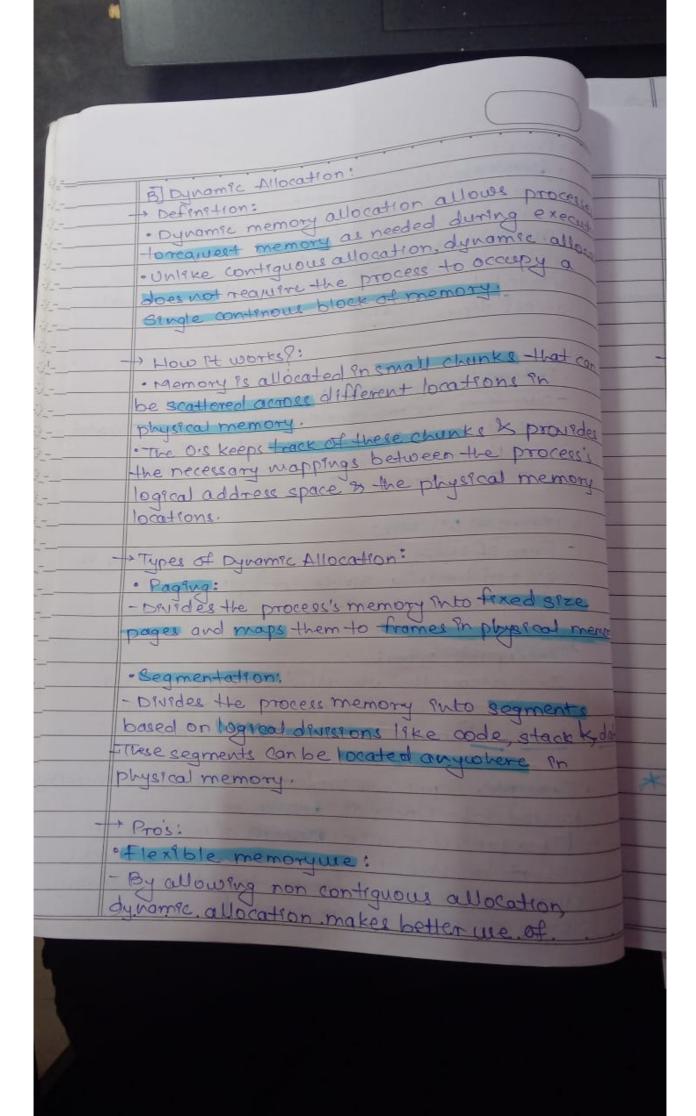


MOMO 1) Paging is a memory management scheme a. 1)
-teat of smarrates the need for contiguous O-8) Paging allocation by) to fixed Size page allocation by bdeviding processes into fixed size page 99) It avoids ext. frag & supports the use of 0.12 uprival memory. Page tables are wed to keep track of where * Hardware reaguered: pages are stored in physical memor * Translation 100 kassale buffers - A cache wed to reduce the time taken to access the page table, speeding up memory access in a paging system. Q.a) Derty but - + It is a bit associated with block of memory that indicates whether the block has been modified + If durty but as set, the block needs to be worsten back to storage (disk) before of can be replaced or removed, 20:10) Shared pages pages of memory, often used for shared Phranels or code segments.

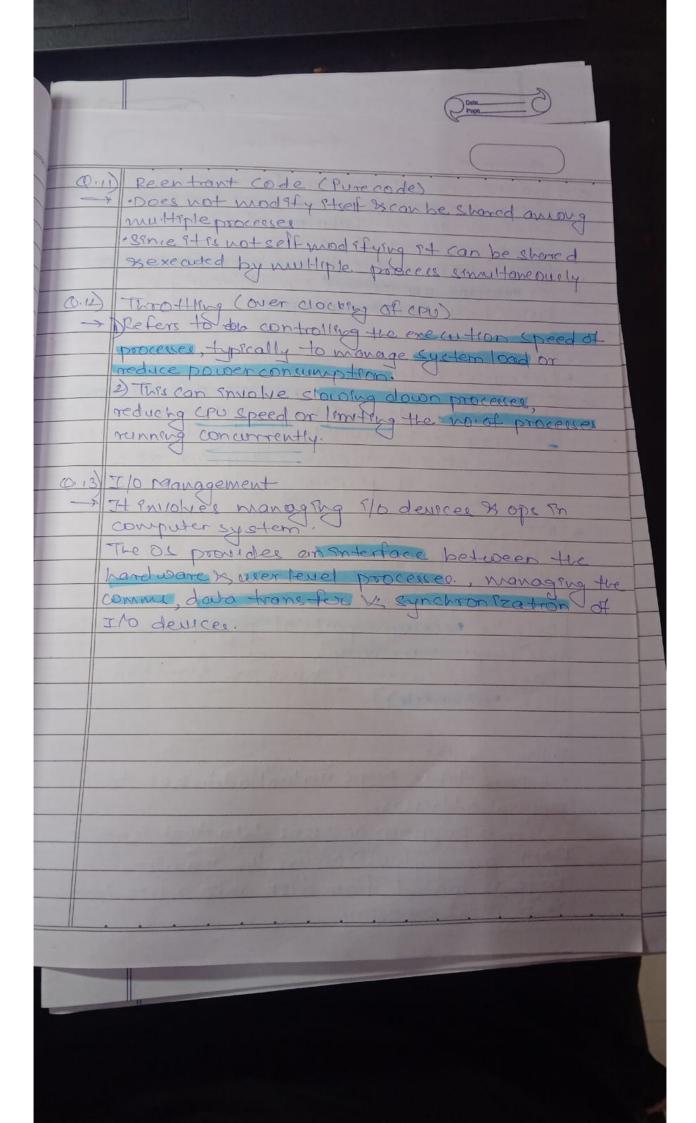


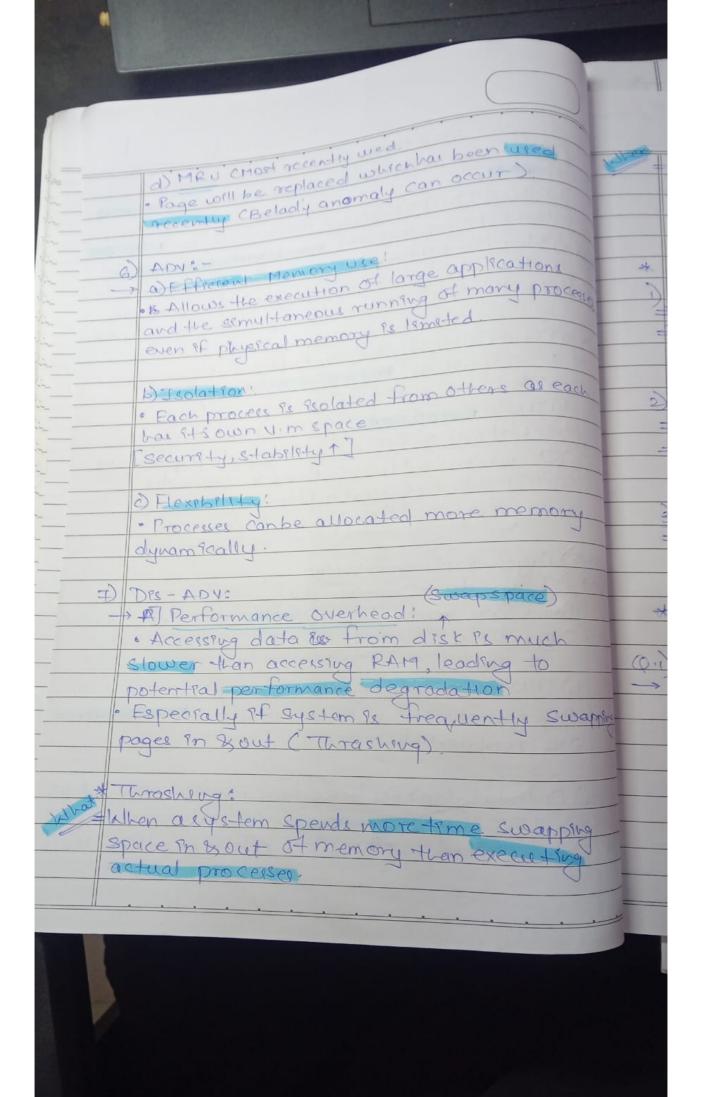
Commayeff These are memory allocation strate for placing Qui First Fet, Best Fet, Worst Fet processes into memory Anocalce the first block of memory that is large on - Pros: Poderce wasted space · Cons: conteded to fragmentation Allocates the smallest block of memory that Best fit: is large enough for the process allocation · Pro: Reduces wasted space · Con: Carlead to ext. fragmentation Delower Elklore+ fet: - Allocates the largest block of memory · Pro: Leaves larger chunks of memory tor future allocation · Con: contead to preffectent memory wage & fragmentation O.D compaction: 1 DIt is a technique used to eliminate ext Fragmentation by monerg all allocated memory blocks to one end of the themony space, combining all tree memory into one large block 90) DAS: It is time consuming process and feasible in all Dis, especially those requiring real - + 8me performance

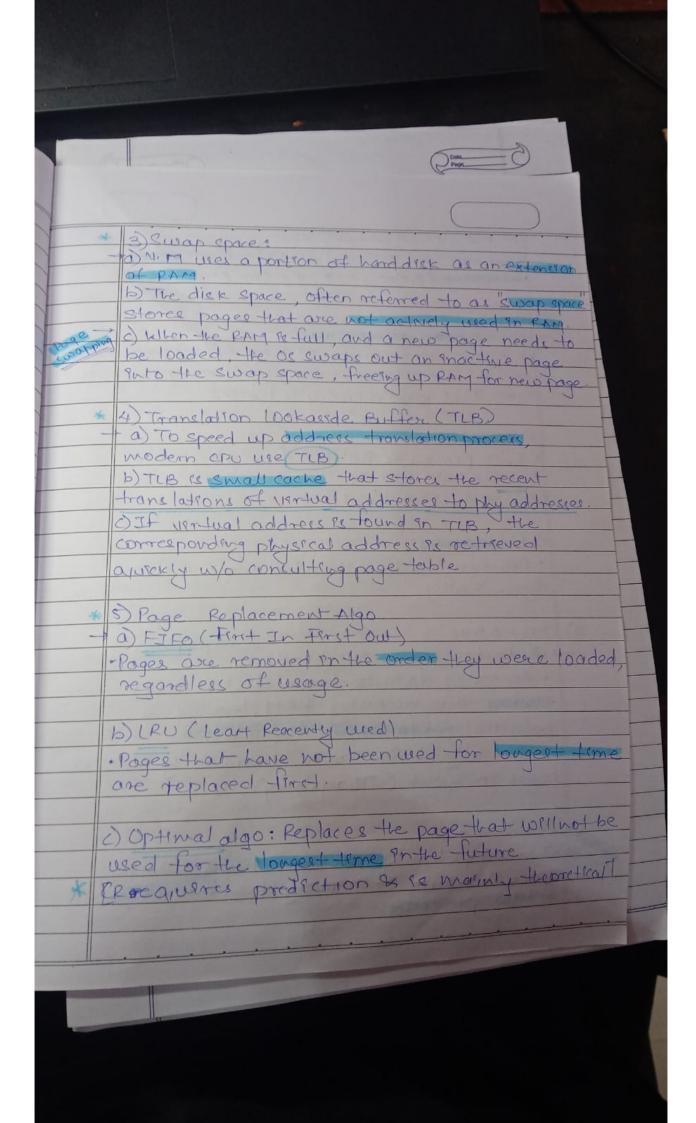
Down C available memory, reducing both internal & · Effecient we of memory: 100 - Memory can be allocated & deallocated as needed, leading to more effectent use of eyetem + con's: · complexety: - bynamic allocation requires the operating system to manage mappings between loggod & physical memory, Increasing complexity · Overhead: keeping track of scattered memory allocation & ensuring effectent access can introduce overhead Example: -If a procees requires BOOMB of memory & the system has three non-configuous block sof 100413 each anallable, the process can still be allocated the required memory by dynamically allocating these non-contiquous blocks. + Comparison: A Contiguous allocation BI Dynamic allocation 1) Sample & fact Dragge complex (2) prone to fragmentation 2) Reduces fragmentation 3) Less flexple 3) 190re flexible (sophisticated memory) (potented memory)

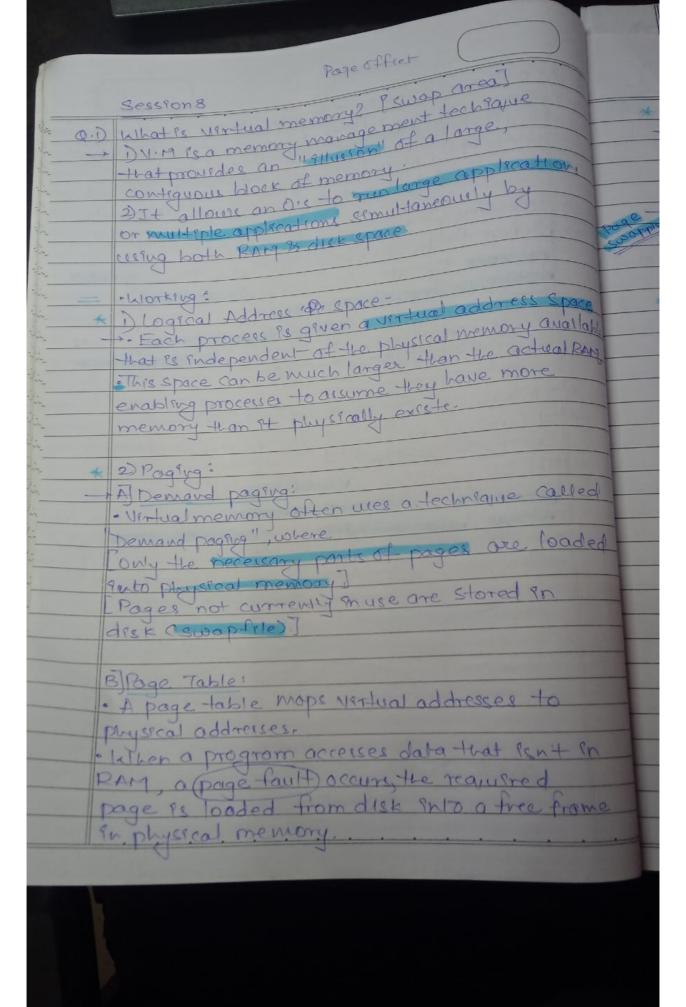


S= 0 Pro's: · Simplicity: Accessing memory is straight forward because the entire process is stored in a erryle, continuous block. · Efficient Memory Access: Since the entire process is stored contiquously there are fewer overheads related to memory access, resulting in factor accels times. - Cons: . External fragmentation: - over time, as processes are allocated & deallocated memory can be come fragmented making it difficult to find a large enough contiguous block for a new process, · Memory Wartage: - Even of there is enough total free memory of may not be in a single contiquous block leading to inefficient memory use. · Fred partitioning: - In systems that we fixed sized partitions, a process might be allocated more memory than It needs, leading to internal fragmentation. · Example: Suppose a sys has GOOMB of free memor divided into blocks of 200MB each. If a process requires 300mB of memory 9+ will not be able to fit into any of these blacks even though there is enough total free meniory as the blocks are not contiquous.









Deadlock prevention & another teast one of the (P.) Deadlock prevention & avoidance four conditions cannot hold & deadlock avoldance which requires making resource allocation decisions dynamically Eyetem never enters a deadlocked state -> A senaphore lea synchronization tool used to Notes (0.3) Semaphore nontrol access to a common resource in a 2) It's an Porteger Marlable that can be we to solve emilical section problems. 8 to ensure mutual exclusion -> A MUTEX is a more restricted type of D.4) MUTEX! semaphore. It is used to lock a resource so that only on thread or process can accels it at a time ensuring mutual exchersion O.D Producer - Consumer Problem DIt is a classic sync problem where two processes the producer & the consumer, share a common buffer 2) The producer adds Heme to the buffer & the consumer removes them. 3) Sewaphores or muteres are typically used to manage access to the buffer & ensure that the producer doein't overflow the buffer & the consumer doesn't try to consume an item when buffer is empty.

