b) Process Isolation:
→ Protects each process from interfering with the memory space of another process.

c) Multitasking:
→ Allows multiple processes to run simultaneously by managing memory allocation & de allocation.

d) Virtual Memory:
→ Enables the execution of processes that require more memory than what is physically available.

Q.3) Contiguous & Dynamic Allocation in M.M
→ A] Contiguous Allocation:
→ Definition:
• Contiguous allocation is a M.M technique where each process is allocated a single contiguous block of memory.
• The entire process's memory needs are satisfied within one continous section of physical memory.

→ How it works?:
• When a process is to be loaded into memory, the O.S searches for a contiguous block of free memory that is large enough to accomodate the process.
• Once found, the entire process is loaded into this block.

Session 7

Q.1) Different types of memories

→ a) Primary Memory (RAM):

→ The main memory where the O.S, applⁿ & data in current use are kept so they can be quickly reached by the device's processor.

b) Secondary Memory (HDD/SSD):

→ Non-volatile memory used for storing data long term, including the O.S, applⁿ & files.

c) Cache Memory:

→ A small sized type of volatile computer memory that provides high speed data access to the processor & stores frequently used computer programs, applications and data.

d) Virtual Memory:

→ An extension of primary memory that provides an "illusion" of a very large main memory by using secondary storage.

e) Registers:

→ Small amounts of fast storage within CPU that store instructions and data currently being used.

Q.2) Need for memory allocation

→ a) Efficient memory utilization

→ Ensures that memory is used efficiently by allocating & deallocating memory as needed.

Momo

Q.8) Paging

→ i) Paging is a memory management scheme that eliminates the need for contiguous allocation by ⇒

→ dividing ~~the~~ memory into fixed size pages

→ dividing processes into fixed size page frames

ii) It avoids ext. frag & supports the use of virtual memory.

* Hardware required :

→ Page tables are used to keep track of where pages are stored in physical memory.

* Translation lookaside buffer:

→ A cache used to reduce the time taken to access the page table, speeding up memory access in a paging system.

Q.9) Dirty bit

→ - It is a bit associated with block of memory that indicates whether the block has been modified.

- If dirty bit is set, the block needs to be written back to storage (disk) before it can be replaced or removed.

Q.10) Shared pages

→ Allows multiple to share the same physical pages of memory, often used for shared libraries or code segments.

Q.6) Internal vs Ext fragmentation.

→ A] Internal frag

→ occurs when fixed sized memory block is allocated & the process does not use all of the allocated memory.

→ The unused space within the block is wasted.

B] Ext. Frag

→ Occurs when free memory is scattered in small blocks throughout the system, making it difficult to allocate large contiguous memory spaces, even if sufficient total memory is available.

Q.7) Segmentation

→ • Segmentation is memory management scheme that supports the user view of memory.

• A program is divided into segments, each of different length.

* Hardware req:

- Requires a segment table that maps logical addresses to physical addresses.
- Each table entry has a base address & limit

* Segmentation Table:

- Contains the base address [starting address of seg in phy memory]
- the limit [limit of the segment]

Convayeff

Q.4] First Fit, Best Fit, Worst Fit
→ • These are memory allocation strats for placing processes into memory.

* a] First Fit:
- Allocates the first block of memory that is large enough for the process.
  - Pro's: ~~Reduces wasted space~~ fast allocation
  - Con's: can lead to fragmentation.

* b] Best fit:
- Allocates the smallest block of memory that is large enough for the process.
  - Pro: Reduces wasted space.
  - Con: Can lead to ext. fragmentation & slower allocation

* c] Worst fit:
- Allocates the largest block of memory.
  - Pro: Leaves larger chunks of memory for future allocation.
  - Con: Can lead to inefficient memory usage & fragmentation

Q.5) Compaction:
→ i) It is a technique used to eliminate ext fragmentation by moving all allocated memory blocks to one end of the memory space, combining all free memory into one large block
ii) Dis: It is time consuming process and is not feasible in all O.S, especially those requiring real-time performance.

available memory, reducing both internal & external fragmentation.

• Efficient use of memory:
- Memory can be allocated & deallocated as needed, leading to more efficient use of system resources.

→ Con's:
• Complexity:
- Dynamic allocation requires the operating system to manage mappings between logical & physical memory, increasing complexity.

• Overhead:
- keeping track of scattered memory allocation & ensuring efficient access can introduce overhead.

Example: -
If a process requires 300MB of memory & the system has three non-contiguous blocks of 100MB each available, the process can still be allocated the required memory by dynamically allocating these non-contiguous blocks.

* Comparison:

A] Contiguous allocation	B] Dynamic allocation
1) Simple & fast	1) More complex
2) prone to fragmentation	2) Reduces fragmentation
3) Less flexible	3) More flexible
(potential memory wastage).	(sophisticated memory management)

B] Dynamic Allocation:

→ Definition:

- Dynamic memory allocation allows process to request memory as needed during execution.
- Unlike contiguous allocation, dynamic allocation does not require the process to occupy a single continous block of memory.

→ How it works?:

- Memory is allocated in small chunks that can be scattered across different locations in physical memory.
- The O.S keeps track of these chunks & provides the necessary mappings between the process's logical address space & the physical memory locations.

→ Types of Dynamic Allocation:

- Paging:
  - Divides the process's memory into fixed size pages and maps them to frames in physical memory.

- Segmentation:
  - Divides the process memory into segments based on logical divisions like code, stack & data.
  - These segments can be located anywhere in physical memory.

→ Pro's:

- Flexible memory use:
  - By allowing non contiguous allocation, dynamic allocation makes better use of

→ Pro's :

- **Simplicity**: Accessing memory is straightforward because the entire process is stored in a single, continuous block.
- **Efficient Memory Access**: Since the entire process is stored contiguously there are fewer overheads related to memory access, resulting in faster access times.

→ Cons:

- **External fragmentation**:
  - Over time, as processes are allocated & deallocated, memory can become fragmented, making it difficult to find a large enough contiguous block for a new process.

- **Memory Wastage**:
  - Even if there is enough total free memory, it may not be in a single contiguous block, leading to inefficient memory use.

- **Fixed partitioning**:
  - In systems that use fixed sized partitions, a process might be allocated more memory than it needs, leading to internal fragmentation.

- **Example**: Suppose a sys has 600MB of free memor divided into blocks of 200MB each.

→ If a process requires 300MB of memory it will not be able to fit into any of these blocks even though there is enough total free memory, as the blocks are not contiguous.

Q.11) Reentrant code (Pure code)
→ • Does not modify itself & can be shared among multiple processes
• Since it is not self modifying it can be shared & executed by multiple process simultaneously

Q.12) Throttling (over clocking of CPU)
→ 1) Refers to ~~the~~ controlling the execution speed of processes, typically to manage system load or reduce power consumption.
2) This can involve slowing down processes, reducing CPU speed or limiting the no. of processes running concurrently.

Q.13) I/O Managgement
→ It involves managing i/o devices & ops in computer system.
The OS provides an interface between the hardware & user level processes, managing the commu, data transfer & synchronization of I/O devices.

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).

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

* 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

* [Requires prediction & is mainly theoretical]

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 application or multiple applications simultaneously by using both RAM & disk space.

Page swapping

• 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.

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's 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.

When # 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