Assignment - 03

Q.1 What are the various ways to avoid deadlock?

Ans Deadlock occurs when more processes are waiting indefinitely for resources held by each other.

The various ways to avoid deadlock are:-

1) Deadlock Prevention: In this method we design are system in such a way that at least one of the necessary condition for deadlock never holds. The four necessary conditions for deadlock are -
1) Mutual exclusion.
2) Hold and wait
3) No Preemption

1) Mutual Exclusion - make resource shareable if possible

2) Hold & wait: Require a process to request all resources at once before execution begins.

3) No Preemption :- Allow preemption if a process holds some resources and request another not available.

4) Circular wait :- Impose a total ordering on resource types.

Q. Deadlock Avoidance - Here the system dynamically resources allocation to ensure a safe state is always maintained. It needs prior knowledge about how resources will be requested.

3) Deadlock detection and Recovery
Although, not strictly "avoidance". this is another approach:
- Allow deadlock to occur then detect and recover.
- Detection algorithm check for cycle in resource allocation graph.

Q2 Describe first fit, best fit and worst fit strategies for disk space allocations with their merits and demerits?

One First Fit : In this method the system scans the free space list and allocate the first hole that is large enough to hold the file of process

Merits : Fast allocation - because it stop searching as soon a suitable block is found.

Demerits : May cause fragmentation a front of memory

Best Fit : Allocate the smallest block that fits the req

Merits : Reduces leftover piece → less waste

Demerits : Slower search time and leads to many small unusable holes.

Worst Fit : Allocate the largest available block.

Merits : Leaves large usable gap still available

Roll No. ..... Scholar No. .....

Demerits : Creates very small useless fragments and may create space

Example:

Free blocks : 20KB, 10KB, 30KB, 25KB

Request: 12KB

- First Fit → 20KB
- Best Fit → 20KB
- Worst Fit → 30KB.

What is fragmentation? Why is it Need Types?

Fragmentation happens when memory is divided such that free space is broken into small pieces, making it difficult to allocate large blocks even when enough total free spe exists.

Fragmentation occurs due to allo and deallocation patterns in memory management

Types of Fragmentation

Type	Meaning	Example
Internal Fragmentation	Wasted space inside allocated blocks due to fixed block allocation	Request 10KB but block size is 20KB → 8KB wasted
External Fragmentation	Free memory exists but as scattered holes, not continuous	Free memory: 5KB + 10KB + 7KB (total 22KB) but request is 20KB: cannot allocate

Q-4 Explain paging and segmentation. How they are helpful in removing fragmentation?

Ans Paging

- Memory is divided into fixed size blocks: frames (physical) and pages (logical)
- Process pages can be loaded anywhere in memory (no contiguous space needed)

Removes External Fragmentation (because any free frame can be used), but can still have internal fragmentation if last page is not fully used

Roll No. ..... Scholar No. .....

Segmentation

- Memory is divided based on logical program units: code, stack, heap, arrays, functions.
- Segments are variable sized and allocated contiguously.

Reduces internal fragmentation because size matches program units.
But may cause external fragmentation because segment sizes vary.

How they help

Feature	Paging	Segmentation
Removes external	Yes	No
Internal	No	Yes (no internal fragmentation)

5. Explain paged segmentation with its hardware implementation.

The technique combines paging + segmentation to get benefits of both methods.

Concept

- Program is divided into segments

- Each segment is further divided into pages
- Removes need for contiguous segment allocation → avoids external fragmentation.

Advantages

- Logical view of segmentation + physical
- No external fragmentation.
- Less external waste than simple paging

Hardware Implementation

Components required:

1. segment table
   - Each entry stores
     - Base address of page table that segment
     - Length of segment.

Roll No. ..... Scholar No. .....

Address Translation steps

Logical Address = Segment Number (s), Page Number (p), offset (d)

Steps:

1. Use segment number s to access the segment table.
2. From the segment table, get the base address of page table for segment s
3. Else page number p to index into that page table → get frame numb
4. Combine frame no + offset d to form physical address.