5.2 Virtual Memory - Introduction to Paging, Segmentation, Fragmentation, and Page fault.

5.2 Virtual Memory

Virtual Memory is a memory management technique that allows the execution of processes that may not be completely in the main memory (RAM). This enables the system to handle larger applications or multiple processes simultaneously by using disk space as an extension of RAM.

Key Concepts

1. Paging

- **Definition:** Paging is a technique used to divide the virtual memory into fixed-size blocks called *pages* and the physical memory into blocks of the same size called *frames*.
- **How it Works:** When a process is executed, its pages are loaded into available frames in physical memory. If a page is not in memory, a page fault occurs, and it is loaded from disk.
- **Example:** If the page size is 4KB, and a process requires 12KB, it will be divided into three pages.
- **Pros:** Eliminates external fragmentation, allows efficient memory allocation.
- **Cons:** May cause internal fragmentation if pages are not fully used.

2. Segmentation

- **Definition:** Segmentation divides the memory into variable-sized segments based on logical divisions such as functions, arrays, or modules.
- **How it Works:** Each segment is allocated to a process, and its size is determined by the size of the logical unit.
- **Example:** A program with three functions may have three segments: Code, Data, and Stack segments.
- **Pros:** Provides logical memory division, simplifies memory management for programmers.
- **Cons:** Can cause external fragmentation.

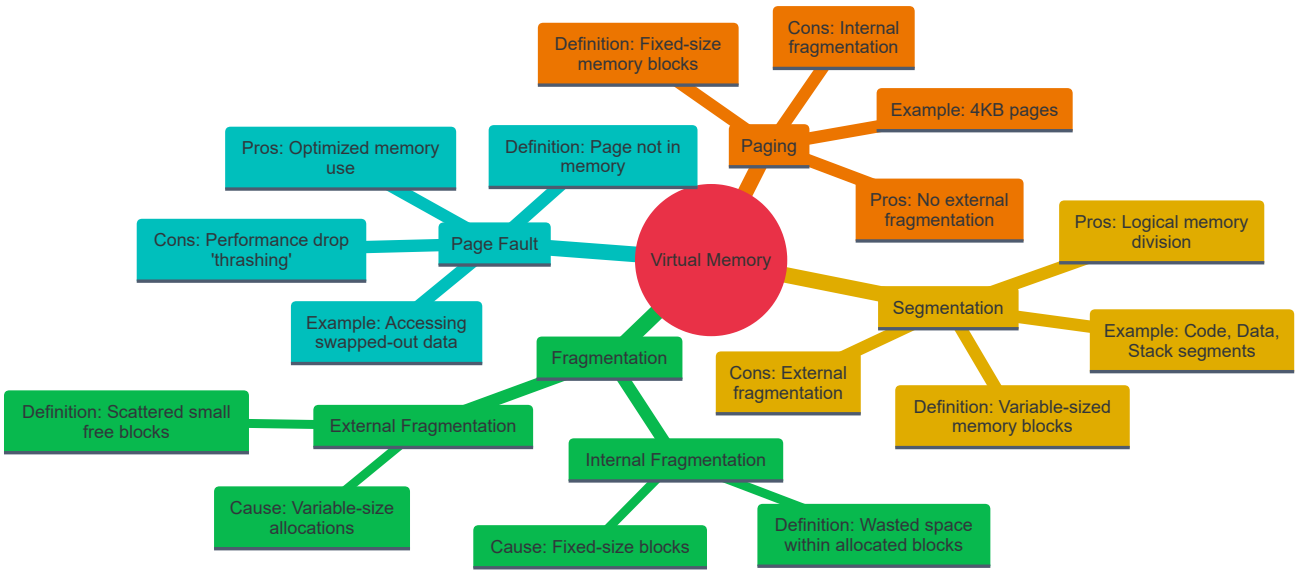
3. Fragmentation

- **Definition:** Fragmentation refers to the unusable spaces left in the memory after allocation.
  - **Internal Fragmentation:** Wasted memory inside allocated blocks due to fixed-size pages or partitions.
  - **External Fragmentation:** Scattered free memory blocks not large enough for new allocations.
- **How it Occurs:** Internal fragmentation occurs when the memory block allocated to a process is larger than required. External fragmentation happens when free memory is divided into small, non-contiguous spaces.

4. Page Fault

- **Definition:** A Page Fault occurs when a program tries to access a page that is not currently in the physical memory.
- **How it Works:** The operating system interrupts the process, retrieves the missing page from the disk, and loads it into memory. This process may involve replacing an existing page in memory.
- **Example:** When a program accesses data on a page that has been swapped out to the disk, a page fault triggers a disk read to bring it back into memory.
- **Pros:** Efficient use of memory by loading only required pages.
- **Cons:** Frequent page faults can lead to high disk I/O, reducing system performance (known as "thrashing").

Diagram: Virtual Memory Concepts



Summary Table of Virtual Memory Concepts

Concept	Definition	Pros	Cons	Example
Paging	Divides memory into fixed-size pages	Eliminates external fragmentation	Causes internal fragmentation	Process divided into 4KB pages
Segmentation	Divides memory into variable-size segments	Logical memory division	Can cause external fragmentation	Code, Data, Stack segments
Fragmentation	Unusable spaces left in memory after allocation	None	Wastes memory (internal/external)	Internal (fixed size), External (scattered)
Page Fault	Occurs when a requested page is not in physical memory	Efficient use of available memory	Can lead to performance issues	Accessing data not currently in memory

Explanation in Simple Terms:

- **Virtual Memory** helps the computer to run many programs by pretending it has more memory than it actually does.
- **Paging** chops up memory into equal-sized boxes so it can fit pieces of programs wherever it finds space.
- **Segmentation** cuts memory into different-sized sections for specific tasks like functions in a program.
- **Fragmentation** happens when memory is wasted due to leftover spaces.
- A **Page Fault** is like asking for a book from the library, but it's not on the shelf – someone has to go get it from storage, slowing things down.