What is Virtual Memory? Explain in detail about the methods of implementing virtual memory?

Virtual memory is a memory management technique that allows a process to use more memory than the physical memory available in the system. [Virtual memory creates an illusion of a large and contiguous logical memory space for each process, while in reality the physical memory may be fragmented and limited1](https://www.geeksforgeeks.org/virtual-memory-in-operating-system/)[2](https://www.javatpoint.com/os-virtual-memory)[3](https://www.codecademy.com/resources/blog/virtual-memory/).

Virtual memory works by mapping the logical addresses generated by the process into physical addresses that can access the main memory or the secondary storage. The mapping is done by using a data structure called page table, which stores the correspondence between each page (a fixed-size unit of logical memory) and its frame (a fixed-size unit of physical memory). The page table is maintained by the operating system and resides in the main memory.

However, not all pages of a process need to be present in the main memory at all times. Some pages may be swapped out to the secondary storage (such as disk) when they are not needed or when the main memory is full. This allows the system to accommodate more processes and use the main memory efficiently. The pages that are swapped out are called invalid pages, and the pages that are present in the main memory are called valid pages.

The following diagram shows how virtual memory works:

The steps involved in accessing a logical address using virtual memory are:

* The CPU generates a logical address, which consists of a page number and an offset within the page.
* The MMU (memory management unit) checks if the page number is present in the TLB (translation lookaside buffer), which is a cache that stores the most recently used page table entries. If yes, this is a TLB hit and the MMU obtains the frame number from the TLB entry.
* If no, this is a TLB miss and the MMU accesses the page table in the main memory using the page number as an index. The MMU obtains the frame number from the page table entry and updates the TLB with this entry.
* The MMU also checks if the page is valid or invalid by looking at a bit in the page table entry. If valid, this means that the page is present in the main memory and can be accessed. If invalid, this means that the page is swapped out to the secondary storage and needs to be brought back into the main memory. This causes a page fault, which is handled by the operating system.
* The MMU combines the frame number and the offset to form the physical address, which can access the main memory or trigger a page fault.

There are different methods of implementing virtual memory, such as:

* Paging: Paging is a method of dividing both logical and physical memory into fixed-size units called pages and frames respectively. Paging allows non-contiguous physical memory allocation for each process by mapping each page to a frame using a page table. Paging eliminates external fragmentation but may cause internal fragmentation within each page[1](https://www.geeksforgeeks.org/virtual-memory-in-operating-system/)[2](https://www.javatpoint.com/os-virtual-memory)[3](https://www.codecademy.com/resources/blog/virtual-memory/).
* Segmentation: Segmentation is a method of dividing both logical and physical memory into variable-size units called segments. Segmentation allows each process to have multiple logical address spaces, such as code, data, stack, etc., which can be mapped to different segments of physical memory using a segment table. Segmentation supports protection and sharing but may cause external fragmentation within each segment[1](https://www.geeksforgeeks.org/virtual-memory-in-operating-system/)[2](https://www.javatpoint.com/os-virtual-memory)[3](https://www.codecademy.com/resources/blog/virtual-memory/).
* Segmented paging: Segmented paging is a method of combining both paging and segmentation techniques. Segmented paging divides both logical and physical memory into fixed-size units called pages and frames respectively. However, each process can have multiple logical address spaces (segments), which are further divided into pages. Each segment has its own page table, which maps its pages to frames. A segment table maps each segment to its corresponding page table. [Segmented paging supports protection, sharing, and non-contiguous allocation but adds complexity to address translation](https://www.geeksforgeeks.org/virtual-memory-in-operating-system/),.