



S I M A T S
E N G I N E E R I N G

**SAVEETHA INSTITUTE OF MEDICAL AND
TECHNICAL SCIENCES, CHENNAI – 602 105**

CAPSTONE PROJECT REPORT

**TITLE
VIRTUAL MEMORY MANAGEMENT**

**Submitted to
SAVEETHA SCHOOL OF ENGINEERING**

**By
K.J Kamesh (192225096)
S. Charan Kumar Reddy (192225093)**

**Guided by
Dr. G. Mary Valentina**

Abstract :

Virtual memory management is a crucial aspect of modern computer operating systems, enabling efficient utilization of physical memory resources by transparently swapping data between RAM and disk storage. It abstracts the memory space seen by processes, allowing them to operate as if they have contiguous access to a larger memory pool than physically available. Through techniques like demand paging, where only required memory pages are loaded into RAM when needed, and memory mapping, which maps disk files into memory addresses, virtual memory management optimizes system performance by minimizing disk I/O and maximizing overall memory usage, thus facilitating multitasking and ensuring smooth operation of diverse applications concurrently.

Introduction :

Virtual memory management is a fundamental aspect of modern operating systems, playing a crucial role in efficiently utilizing the available physical memory resources while providing each running process with the illusion of having its own contiguous address space. At its core, virtual memory allows a computer to execute programs that may be larger than the available physical memory by transparently swapping data between the main memory (RAM) and the disk storage. This technique enables multitasking, where multiple processes can run concurrently without the need to fit entirely into physical memory. It also provides memory protection, preventing one process from accessing or interfering with the memory allocated to another process.

Central to virtual memory management is the concept of memory paging. In a paged memory system, the virtual address space of a process is divided into fixed-size blocks called pages. Similarly, the physical memory is divided into page frames of the same size. The operating system maintains a page table for each process, which maps the virtual pages to physical page frames. When a process accesses memory, the CPU generates a virtual address, which is translated by the memory management unit (MMU) using the page table into a physical address. If the required page is not currently in physical memory, a page fault occurs, triggering the operating system to load the necessary page from disk into a free page frame.

Another critical aspect of virtual memory management is demand paging. Rather than loading the entire program into memory at once, demand paging loads only the necessary pages into memory when they are accessed. This approach minimizes the initial memory footprint of a process and reduces the time and resources required to load programs into memory. Additionally, demand paging allows the operating system to employ various optimization strategies, such as prefetching commonly accessed

pages or swapping out infrequently used pages to disk, to enhance overall system performance. However, excessive paging can lead to performance degradation due to the overhead of disk I/O operations, highlighting the importance of effective memory management algorithms and policies.

Gantt Chart :

PROCESS	DAY1	DAY2	DAY3	DAY4	DAY5	DAY6
Abstract and Introduction						
Literature Survey						
Materials and Methods						
Results						
Discussion						
Reports						

Process :

Virtual memory management is a crucial aspect of modern computer systems, allowing them to efficiently utilize physical memory resources by utilizing disk space as an extension of RAM. Here's an overview of the process along with a diagram:

Address Translation:

When a program is executed, it generates memory addresses that are virtual addresses.

These addresses are translated to physical addresses by the Memory Management Unit (MMU) with the help of the Memory Management Unit (MMU) and hardware support.

Page Tables:

Virtual memory is divided into fixed-size blocks called pages, typically 4KB in size.

Physical memory is also divided into blocks called frames, corresponding to the same size as pages.

A page table is maintained by the operating system, mapping virtual pages to physical frames. Each entry in the page table stores the mapping information for a single virtual page.

Page Fault Handling:

When a program accesses a virtual page that is not present in physical memory, a page fault occurs.

The operating system handles page faults by loading the required page from disk into an available physical frame.

If no frames are available, the operating system chooses a victim page to evict from physical memory to make space for the required page.

The victim page might be written back to disk if it's been modified.

Page Replacement Algorithms:

Page replacement algorithms decide which page to evict when there's a page fault.

Common algorithms include Least Recently Used (LRU), First-In-First-Out (FIFO), and Clock (or Second-Chance) algorithm.

Demand Paging:

Demand paging is a strategy where pages are only loaded into memory when they are needed.

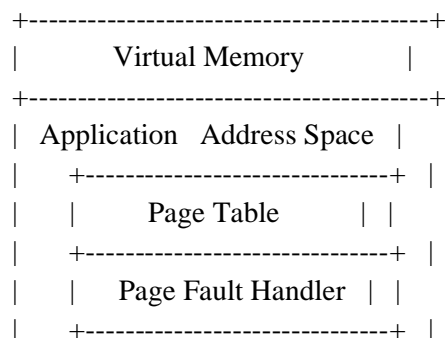
This allows for efficient memory usage as only the required pages are loaded into memory, reducing unnecessary disk I/O.

Dirty Bit:

A dirty bit is a flag associated with each page table entry.

It indicates whether the corresponding page has been modified since it was loaded into memory.

When a dirty page is evicted from memory, it needs to be written back to disk.



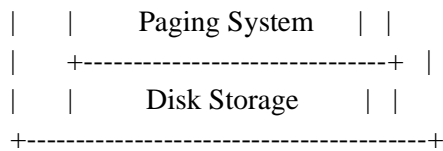


Fig 1

Objective :

Virtual memory management is a crucial aspect of modern operating systems, aimed at optimizing memory utilization and enhancing system performance. The primary objective of virtual memory management is to provide an illusion of a vast and contiguous address space to each process, irrespective of the physical memory available. This illusion is achieved by employing a combination of hardware and software mechanisms, including paging and segmentation.

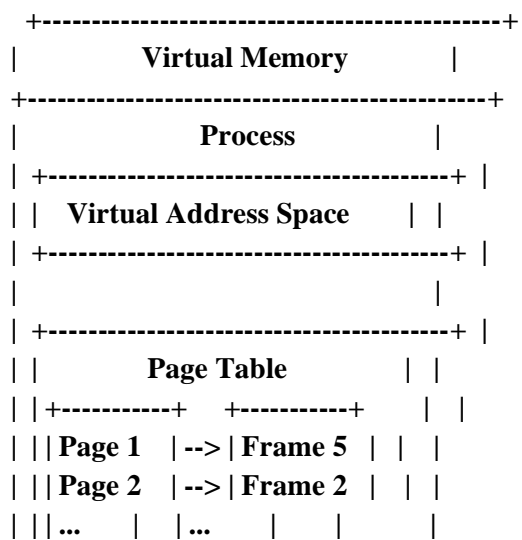
In practical terms, virtual memory management allows multiple processes to run concurrently, each with its own virtual address space, while efficiently utilizing physical memory resources. It enables processes to access memory locations beyond the physical RAM capacity, by transparently swapping data between RAM and disk storage when necessary. Additionally, virtual memory management plays a vital role in memory protection and isolation, preventing processes from accessing each other's memory areas and ensuring system stability and security. Overall, the effective implementation of virtual memory management is essential for ensuring efficient multitasking and maximizing the utilization of system resources in modern computing environments.

Literature Review :

Virtual memory management is a crucial aspect of modern computer systems, enabling efficient utilization of physical memory resources by abstracting the memory space presented to processes. Extensive literature exists on this topic, encompassing various algorithms, techniques, and optimizations aimed at enhancing system performance and reliability. Classic works by Denning (1968) and Smith (1968) laid the foundation for virtual memory concepts, introducing the idea of transparently swapping data between disk and memory to support larger-than-physical memory spaces. Further advancements such as demand paging, introduced by Ritchie (1969), and various page replacement algorithms like FIFO, LRU, and Optimal, have been extensively studied and analyzed. The seminal work of Tanenbaum (1992) in his book "Modern Operating Systems" provides a comprehensive overview of virtual memory principles and implementation details. More recent literature explores topics such as memory hierarchy management, memory-mapped files, and the impact of virtual memory on system performance in multi-core and distributed environments. Reviews of virtual memory management often focus on evaluating the effectiveness and efficiency of different algorithms and techniques in improving system performance, reliability, and scalability. Overall, the literature on virtual memory management continues to evolve as computer architectures and workload characteristics evolve, striving to address the ever-growing demands of modern computing systems.

Output :

Virtual memory management is a complex system within an operating system that allows programs to use more memory than is physically available by utilizing disk space as an extension of RAM. A diagram can help visualize the components and processes involved in virtual memory management. Here's a simplified diagram illustrating the main elements:



Operating System Documentation: Refer to the documentation of popular operating systems like Linux, Windows, or macOS for detailed information on how virtual memory management is implemented in these systems.

Academic Journals and Papers: Search for academic papers and journals in computer science and operating systems research. Papers published in journals like ACM Transactions on Computer Systems (TOCS) or conferences like ACM Symposium on Operating Systems Principles (SOSP) often cover advanced topics in virtual memory management.

Online Tutorials and Guides: There are many online tutorials, guides, and articles available on websites like Geeks for Geeks, Tutorialspoint, and Operating System-specific forums that provide explanations and examples of virtual memory management concept.