Operating System week 4

Khaula Molapo

1.

In a cloud platform, virtual memory allows multiple users to run applications without worrying about the physical memory limits of the server. For example, imagine a virtual server hosting several web applications. Each application is assigned its own virtual address space, which the operating system maps to physical RAM when needed. If the RAM is full, less-used data is temporarily moved to disk storage, creating a swap space. This ensures that applications continue running smoothly even when demand spikes. Virtual memory also isolates programs from each other, preventing one program from crashing the entire system. In cloud platforms, this setup allows servers to efficiently manage multiple workloads, scale resources dynamically, and maintain performance. Users don't see these memory transfers directly, but the system ensures data is quickly accessible. Virtual memory is essential for handling large datasets, multitasking, and providing a seamless user experience in modern cloud environments.

2.

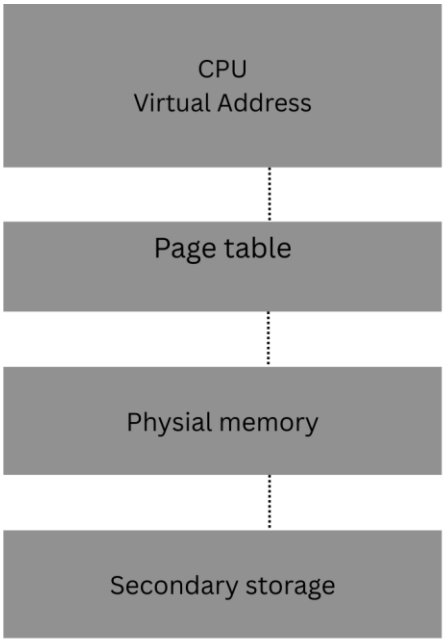
The Least Recently Used (LRU) algorithm is a page replacement strategy used in virtual memory management. When memory is full and a new page needs space, LRU removes the page that hasn't been used for the longest time. The idea is that pages used recently are more likely to be needed again soon. LRU helps improve system efficiency by minimizing the number of page faults, which occur when required data isn't in RAM. Operating systems implement LRU using counters, stacks, or linked lists to track page usage. It is widely used in caching, databases, and cloud platforms to optimize memory utilization and performance.

3.

Reflection: This program simulates a simple memory with limited slots and shows how pages are replaced using a basic FIFO-like approach. Writing it in Visual Studio Code helped me understand how memory slots are updated dynamically. Testing the simulation highlighted how virtual memory concepts, like page replacement, work in practice. I noticed the importance of tracking page usage to optimize memory efficiency. Visualizing memory changes step-by-step reinforced my grasp of memory management and replacement strategies. It was a practical exercise connecting theory to coding, showing how operating systems handle limited memory in real scenarios.

4.

The memory diagram shows how virtual memory works by separating virtual addresses from physical RAM. Virtual addresses from programs are mapped to physical memory through a page table. Frequently used data stays in RAM, while less-used pages move to disk swap space. The diagram includes CPU, virtual memory, page table, RAM, and disk storage. Arrows indicate data movement between virtual memory and physical memory, illustrating page fetching and replacement. This visual representation helps understand memory isolation, efficient multitasking, and how systems handle memory overflow. Virtual memory diagrams make it easier to see how operating systems manage data access and performance.


CPU
Virtual Address
Page table
Physial memory
Secondary storage