Memory virtualization is a technique used in computer systems to provide an abstraction layer between the physical memory (RAM) and the virtual memory used by processes or programs running on the system.

It allows each process to have the illusion of a large, contiguous address space, even if the physical memory is limited.

Here’s an overview of how memory virtualization works:

1. Virtual Memory Space: Each process is assigned a virtual memory space, which is typically much larger than the available physical memory. This virtual address space is divided into fixed-size units called pages.

2. Page Tables: The operating system maintains a data structure called the page table for each process. The page table maps the virtual addresses used by the process to physical addresses in the physical memory.

3. Page Faults: When a process accesses a virtual address that is not currently mapped to a physical address, a page fault occurs. This can happen when the required page is not present in the physical memory.

4. Page Replacement: In the case of a page fault, the operating system selects a page to evict from the physical memory to make space for the required page. This is known as page replacement. The evicted page is written back to disk if it was modified.

5. Disk Paging: When a page needs to be evicted from the physical memory, it may be temporarily stored on disk in a designated area called the swap space. This allows the operating system to free up physical memory by moving less frequently used pages to disk.

6. Page Table Updates: When a page is replaced or moved to disk, the corresponding entry in the page table is updated to reflect the new physical address or indicate that the page is on disk.

7. Demand Paging: Virtual memory systems often employ demand paging, which means that pages are loaded into the physical memory only when they are actually needed. This helps to optimize memory usage by bringing in pages on-demand rather than loading the entire process into memory at once.

Memory virtualization provides several benefits, including:

1. Increased Address Space: It allows processes to utilize a larger address space than what is physically available, enabling the execution of larger programs or multiple programs concurrently.

2. Memory Isolation: Each process has its own virtual memory space, which provides memory isolation and protects processes from interfering with each other's memory.

3. Simplified Memory Management: Memory virtualization simplifies memory management for both the operating system and application developers. It abstracts away the details of physical memory allocation and allows for more flexible memory usage.

4. Efficient Memory Utilization: By using techniques like demand paging and page replacement, memory virtualization optimizes the usage of physical memory by keeping frequently accessed pages in memory and swapping out less used pages to disk.

Related posts:

1. Information Life Cycle Management (ILM)
2. Storage infrastructure
3. Integrated VS Modular Array
4. Data proliferation
5. Data categorization
6. Component architecture of intelligent disk subsystem
7. Intelligent disk subsystems overview
8. Mapping n operations
9. Storage system architecture
10. RAID
11. Hot spare
12. SAN security
13. JBOD
14. Elements of DAS,NAS,CAS,SAS
15. Limitations of DAS
16. Cloud vocabulary
17. NAS security
18. Management of DAS,NAS,CAS,SAN
19. FC Connectivity
20. Data center concepts & requirements
21. Network virtualization
22. Server information storage and management
23. ISM Architectural Framework