Name: Vin Sovichhay

ID: e20241261

Group: I2-AMS2B

TP4

1. What is virtual memory? How many main types of virtual memory are there?

+ Virtual memory is a technique used by the operating system to manage memory more efficiently by extending the available memory using the hard disk. It allows programs to run even if they are larger than the actual RAM by temporarily moving some data between RAM and secondary storage. This creates the illusion that the system has more memory than it physically has.
+ There are two main types of virtual memory:
    - Paging: memory is divided into equal-sized blocks called pages.
    - Segmentation: memory is divided into different-sized parts based on the structure of the program.

2. Why is virtual memory important? Please be sure to explain more details.

+ Virtual memory is important because it helps a computer use its memory more efficiently and handle situations when the physical RAM is not enough. Instead of stopping programs when memory runs out, the operating system uses part of the storage (like a hard drive or SSD) as extra memory. This allows the system to continue working smoothly.
+ One key reason virtual memory matters is that it allows a computer to run multiple programs at the same time. Even if the RAM is limited, the system can move less-used data to disk and free up space for active tasks. This makes multitasking possible and improves overall system performance.
+ It is also important because it enables computers to run larger applications that normally would not fit into physical memory. Programs can use more memory than what is actually installed, which is useful for complex software and modern applications.
+ Another benefit is that virtual memory helps optimize memory usage automatically. The system decides which data should stay in RAM and which can be moved to storage, reducing the need for users to manually manage memory or upgrade hardware immediately
+ However, virtual memory is slower than RAM, so if the system relies on it too much, performance can decrease. Despite this limitation, it is still essential because it prevents programs from crashing and ensures the computer can continue operating even when memory is limited.

3. What is paging? Explain the process of paging.

+ Paging is a memory management technique used by an operating system where a program is divided into small fixed-sized parts called pages. These pages are stored in secondary memory and loaded into main memory (RAM) when needed. This allows a program to be stored in non-contiguous locations in memory.
+ Process of Paging:
    - First, the operating system divides a program into equal-sized pages.
    - The main memory is also divided into fixed-size blocks called frames.
    - When a program runs, its pages are loaded into any available frames in RAM
    - The system uses a page table to keep track of where each page is stored in memory

- When the CPU needs data, it checks the page table to find the correct frame and access the required information.

4. Explain about “Page fault”. What does it mean? Demonstrate on how to handle page fault.

+ A page fault is a situation that happens when a program tries to access data or instructions that are not currently stored in the main memory (RAM). Instead, the required data is located in secondary storage, so the operating system must load it into RAM before the program can continue.
+ How page fault is handled:
    - The system detects the page fault and temporarily stops the current that the address is valid
    - The operating system checks which page is required and confirms that the address is valid
    - It looks for a free space (frame) in RAM. If none is available, it removes an existing page using a replacement method
    - If the removed page has been modified, it is saved back to the disk
    - The required page is then loaded from the disk into RAM
    - The page table is updated to show the new location of the page
    - Finally, the program resumes execution as if nothing happened.

5. Describe FIFO (First In, First Out) and LRU (Least Recently Used). Explain with example.

+ FIFO (First In, First Out) and LRU (Least Recently Used) are page replacement algorithms used by an operating system to decide which page should be removed from memory when a new page needs to be loaded.
+ FIFO (First In, First Out): is the simplest method. It removes the page that has been in memory the longest (the one that entered first). The system treats pages like a queue, where the oldest page is removed first.
+ Example: Assume we have 3 memory frames and pages: 1, 3, 0, 3, 5
    - 1, 3, 0 : loaded into memory
    - When 5 arrives, memory is full
    - The oldest page (1) is removed
    - So, 5 replaces 1
+ LRU (Least Recently Used): LRU replaces the page that has not been used for the longest time. It keeps track of page usage and removes the least recently accessed page.
+ Example: Assume page: 7, 0, 1, 2, 0, 3 with 3 frames
    - 7, 0, 1 : loaded into memory
    - When 2 arrives : replace 7 (least recently used)
    - When 3 arrives: replace 1 (least recently used)
+ Summary:
    - FIFO: removes the oldest page
    - LRU: removes the least recently used page

References

1. https://www.geeksforgeeks.org/operating-systems/virtual-memory-in-operating-system/
2. https://www.techtarget.com/searchstorage/definition/virtual-memory
3. https://unstop.com/blog/what-is-paging-in-os
4. https://www.geeksforgeeks.org/operating-systems/page-fault-handling-in-operating-system/
5. https://www.geeksforgeeks.org/operating-systems/page-replacement-algorithms-in-operating-systems/