Windows vs Linux: Memory Management Strategies

Shahd Tharwat El-Refai, ID: 8504

December 2024

1 Introduction

Memory is the main part of any operating system - it's constantly changing, adapting, and ensuring no problems happens. This happens with the help of different memory management strategies. Both Windows and Linux approach these challenges in a bit different ways, as we will see Windows is more simple ready to use while Linux provides more flexibility which is more advanced.

2 What is Memory Management?

Memory management is about efficiently organizing the available memory space while preventing any programs from interfering with one another. Some key strategies include:

- **Paging:** A way of retrieving processes from secondary storage to the main memory as pages.
- **Virtual Memory:** A way of mapping chunks of memory to disk files which makes the computer treat secondary memory like a main memory.
- **Swapping:** Moving data between memory and disk storage when memory is needed.

3 Virtual Memory

Virtual memory gives applications the illusion of a large, continuous block of memory, regardless of the actual physical RAM available. This is achieved by mapping virtual addresses to physical ones, with disk space acting as an extension of RAM when needed.

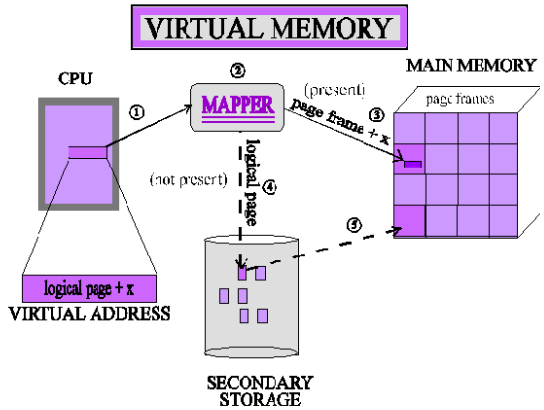

Figure 1: Virtual Memory.

3.1 Windows Virtual Memory

Windows provides each process with its own Virtual Address Space (VAS) that is split between user mode and kernel mode. For 32-bit systems, this split is typically 2GB each, but it can differ on 64-bit systems. Some features include:

- **Memory-Mapped Files:** These allow efficient file access and sharing between processes by mapping files directly into memory.
- **Demand Paging:** Pages are only loaded into RAM when they're accessed which reduces memory usage and also improves efficiency.
- **Pagefile:** A pagefile that is like an extension of physical memory which is dynamically managed. Its size can be configured by the users themselves or Windows manage it automatically.

3.2 Linux Virtual Memory

Linux also assigns each process its own VAS also dividing it into user and kernel spaces. Some features include:

- **Memory-Mapped Files:** It is used for shared libraries and device interactions. This allows processes to share memory regions efficiently.
- **Demand Paging:** Like Windows, it loads pages into memory only when accessed.
- **Swap Space:** Linux uses a dedicated swap partition or file to extend physical memory. Also users can adjust `vm.swappiness` parameter. This parameter controls how swap space is used.
  - **Swappinnes = 0:** Means the kernel will avoid swapping unless absolutely necessary. It favors keeping data in RAM even if memory is tight.
  - **Swappinnes = 100:** Means the kernel will swap aggressively, even when there's still plenty of free memory available.

3.3 Comparison

The biggest difference I see is with swappiness variable which gives more control.

4 Paging

The OS breaks memory into fixed-size blocks, these are pages. It helps with allocation into memory and improves good use of it and minimizing waste. Both Windows and Linux use paging, but their implementation differs.

4.1 Windows Paging Policies

Windows has a hierarchical paging system to map virtual addresses to physical ones efficiently. Its strategies include:

- **Hierarchical Page Table:** Page tables are organized in a tree structure way. It is a specefic way of multi-level.
- **Page Replacement:** Happens with use of a modified Least Recently Used (LRU) algorithm. It decides which pages to keep in memory and what to replace.
- **Prefetching:** Preloads pages that it thinks will be accessed in the future.

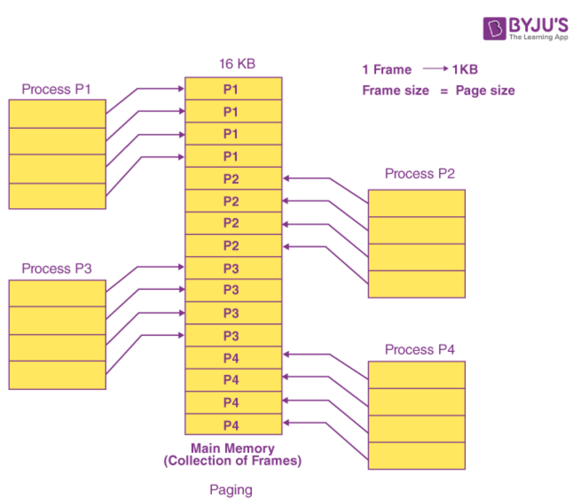

Figure 2: Paging.

4.2 Linux Paging Policies

Linux uses a multi-level page table for address translation and offers some extra tuning options:

- **Tunable Parameters:** Settings like `vm.swappiness` allow users to customize paging behavior.
- **Clock-Pro Algorithm:** this is an improved version of the Clock algorithm. It considers both how recently and frequently a page has been accessed.
- **Huge Pages:** Supports large page sizes this betters performance for applications that require a lot of memory usage.
- **Active and Inactive Lists:** Keeps track of recently and less-recently used pages to better manage resources.

4.3 Comparison

Windows paging system is simple and automated so it is more user-friendly. Linux paging system is more complicated and has tuning options.

5 Swapping

Swapping moves memory pages from RAM to disk storage to free up space for active processes. This happens specially during time of high memory usage.

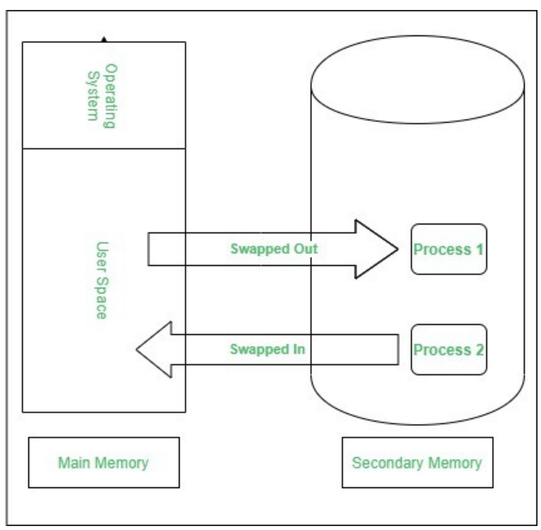

Figure 3: Swapping.

5.1 Windows Swapping Policies

Windows manages swapping through its pagefile:

- **Pagefile Utilization:** Based on usage patterns, Windows automatically decides what to move to pagefile.

- **Priority-Based Swapping:** Active and also more frequently accessef pages stay in RAM and do not get swapped.
- **Automatic Management:** Swapping operations happens without much user input, but users can still adjust the pagefile settings if they want.

5.2 Linux Swapping Policies

Linux uses a dedicated swap partition or file:

- **Swappiness Parameter:** Adjusts how often the system uses swap space as mentioned in virtual memory section.
- **Out-Of-Memory (OOM) Killer:** When swap space has been exhausted this mechanism kill processes to free up memory.
- **Page Eviction Policies:** Algorithms like Clock-Pro help decide which pages to swap out.

5.3 Comparison

Again, Windows focuses on simplicity and automation, while Linux gives users the tools to fine-tune swapping behavior for specific workloads.

6 General Comparison

Memory usage on Windows vs. Linux has distinct performance implications, especially in terms of swapping and paging. Windows uses a unified memory management system that heavily relies on paging, which can lead to performance degradation under high memory pressure due to frequent page file access. In contrast, Linux uses a more flexible system that minimizes swapping and relies on available RAM more efficiently. This approach reduces disk I/O, which can lead to better performance under similar conditions, as Linux is able to manage memory demands with less frequent reliance on swap space.

Both systems use swapping and paging to manage virtual memory beyond the limits of physical RAM, but their implementation differs. In Windows, this is handled through a "paging" file, where inactive memory pages are moved to disk to free up RAM. On Linux, swap space can be a dedicated partition or a swap file, and the choice between these options can significantly impact performance. Excessive swapping, where data is frequently moved between RAM and swap space, can cause disk thrashing and lead to significant performance degradation. However, Linux's ability to use a dedicated swap partition, optimized for faster access, offers more consistent performance under high memory loads compared to Windows' paging system, where performance can suffer due to the limitations of the page file located on a regular filesystem.