9-2 Final Project: Technical Evaluation of an Operating System

Darren J. Oates

Southern New Hampshire University

IT-600-10151-M01 Operating Systems 2025 D-4

Michael Gallaher

October 26th, 2025

Technical Evaluation of an Operating System

Organization Profile

TSI's back-office operating system requires preemptive multitasking and advanced scheduling features for multiprogramming. They can become very busy during peak work hours and employees require multiprocessing on its workstations. TSI has to purchase old, decommissioned hardware with single-core processors for its data center. Back-office applications that offer multithreaded operation hang at launch, so TSI has to use open-source software so that a team of TSI software developers can remove multi-threading functionality. TSI OS uses a flat memory model without paging. To address performance issues in its back office, TSI should adopt a general-purpose operating system with virtual memory support, allowing the system to handle more applications than the physical RAM would normally permit. At TSI headquarters, developers have had to modify traps to kernel mode and develop custom system call responses. Switching to a modern general-purpose OS, Windows 11 would introduce this missing layer, giving TSI's software a reliable way to interact with the system. TSI handles data for government and Fortune 500 clients, which demands high security. TSI's back office has a variety of devices that are large and sophisticated and require optimal device drivers. All company data resides on individual SATA and SCSI drives.

Function, Management, and Maintenance

An effective operating system for TSI must fulfill comprehensive organizational, security, performance, and reliability requirements to support its expanding back-office operations. From an organizational standpoint, the OS must be manageable and scalable, enabling efficient developer workflows through features like a standardized system call interface and preemptive multitasking. This reduces the need for low-level kernel coding and accelerates application development. In terms of security, the OS must enforce strict access controls, audit trails, and encryption protocols to protect sensitive government and enterprise data, replacing TSI's current unsecured systems, where any network user can access critical infrastructure. For performance, the operating system should support multiprogramming, multiprocessing, multithreading, and virtual memory, enabling efficient CPU utilization, parallel processing, and stable multitasking under limited hardware resources. These features reduce application wait times, prevent crashes due to memory shortages, and optimize throughput across diverse workloads. Regarding reliability, the OS must integrate fault-tolerance mechanisms such as RAID support, snapshots, and failover recovery to ensure high availability and business continuity, especially when working with decentralized storage or under hardware failure scenarios. A modern general-purpose operating system (GPOS) that combines these capabilities will eliminate inefficiencies inherent in TSI's embedded OS, enabling the company to match the performance standards it delivers to clients, while ensuring secure, scalable, and resilient internal operations.

Hardware

To support a modern general-purpose operating system (GPOS) capable of meeting TSI's business demands, the hardware must include multi-core CPUs, sufficient RAM, and enterprise-grade storage systems. A multi-core processor is essential for enabling multiprocessing and multithreading, ensuring smooth performance across concurrent applications. At least 16 GB of RAM is recommended to support virtual memory functions and prevent system crashes under heavy multitasking workloads. Solid-state drives (SSDs) with RAID support offer the speed and fault tolerance necessary for handling large files and ensuring data availability. Network interface cards with proper driver support are also required for seamless device communication. TSI's current reliance on single-core, outdated hardware limits scalability and forces inefficiencies that conflict with its organizational need for fast, reliable, and secure operations. Upgrading to modern, GPOS-compatible hardware will ensure the infrastructure can support secure access control, system calls, virtualization, and high-speed multitasking, aligning the back-office environment with the same performance standards expected in its client-facing products.

Support and Functionality

The architecture of a modern general-purpose operating system (GPOS)such as Windows 11 provides comprehensive support for process management, memory management, input/output operations, and mass storage—each critical to TSI's operational efficiency. Its process management system supports preemptive multitasking, multiprocessing, and multithreading, allowing multiple applications to run concurrently with fair CPU scheduling and rapid context switching. Memory management is handled through virtual memory, enabling applications to access more memory than is physically available and preventing system crashes due to RAM limitations. I/O functionality is streamlined through standardized system calls and robust device driver support, allowing seamless communication between software and diverse hardware without the need for custom drivers. Mass storage is managed through fault-tolerant systems like RAID, snapshots, and automated failover, ensuring high availability, data integrity, and quick recovery in case of hardware failure. This architectural foundation allows TSI to scale securely, minimize downtime, and maintain consistent performance under demanding workloads.

Architectural Issues

From TSI's perspective, the architectural support for multiprocessor systems enables parallel processing by organizing multiple CPUs to share system memory, I/O devices, and communication pathways efficiently under coordinated control mechanisms. The architecture's organization relies on symmetric multiprocessing (SMP) or distributed shared-memory configurations, allowing balanced load distribution and fault isolation. Connection topologies, such as crossbar switches or high-speed interconnects, facilitate low-latency data exchange, while control is managed through sophisticated scheduling algorithms and cache coherence protocols that maintain data consistency across processors. However, architectural issues such as memory bottlenecks, cache coherence overhead, and interprocessor communication delays can degrade system performance if not properly addressed. Additionally, the technologies used—such as multicore processors and shared memory buses—introduce complexities in synchronization and scalability as workloads increase. To mitigate these challenges, TSI should implement multiprocessing through a general-purpose operating system (GPOS), Windows 11, designed to optimize CPU scheduling, manage shared memory access, and balance loads dynamically. This approach would maximize hardware utilization, enhance reliability, and support scalable performance, ensuring TSI's infrastructure aligns with enterprise-grade computing standards and future operational growth.

Responsiveness to Organizational Needs

In TSI's back-office, process management in an operating system must act like a well-oiled engine—not just keep things running but keep them moving and responsive. When the OS uses advanced scheduling, preemptive multitasking, and threading support, it keeps applications from stalling and employees from waiting. By using process-monitoring tools such as Windows Task Manager that capture real-time CPU load, memory footprints, and process priority shifts, the IT team can spot the slow spots and fine-tune the system. The result isn't just stable it's agile, efficient, and aligned with the organization's pace.

In modern IT environments, resource management goes hand in hand with process control. An operating system that intelligently allocates memory, disk, and network bandwidth ensures that no single application hoards resources or causes bottlenecks. By dynamically balancing these loads, systems can maintain peak performance even under pressure from multiple users or demanding applications. Administrators rely on diagnostic tools and performance counters to visualize these patterns over time, revealing trends that simple snapshots can't show. This data-driven insight allows teams to anticipate issues before they escalate replacing reactive troubleshooting with proactive optimization that keeps operations seamless.

Equally vital is system compatibility and maintenance, where updates, driver management, and licensing compliance form the foundation of long-term reliability. Keeping operating systems patched and drivers current prevents conflicts, security gaps, and downtime that could ripple across the enterprise. Automated update schedules, license audits, and standardized configurations simplify what was once a tedious manual process. When every OS instance and application is verified, supported, and properly licensed, the organization not only protects itself legally but also guarantees that all digital components speak the same technical language. This disciplined approach transforms IT infrastructure from a collection of systems into a unified, secure, and forward-ready ecosystem.

Software Tools

From TSI's perspective, adopting a general-purpose operating system unlocks access to advanced software tools for thread analysis and deadlock detection, which are essential for maintaining reliable and efficient back-office operations. Tools such as VisualVM, Perf, and Windows Performance Analyzer (Windows-Driver-Content, 2020) allow TSI's developers to trace thread behavior, monitor concurrency issues, and identify resource contention in real time. (Documentation) These tools support proactive strategies like detecting circular wait conditions, logging thread states, and analyzing lock acquisition patterns to prevent or resolve deadlocks before they impact performance. By leveraging built-in OS support for multithreading and these diagnostic tools, TSI can implement effective thread management strategies—such as timeout-based locking, lock hierarchy enforcement, and thread prioritization—that ensure smooth execution across applications, reduce debugging time, and keep the system running at full speed.

Support

TSI should see the operating system as more than just software—it's the engine room that keeps every part of the business moving. In a fast-paced environment where back-office systems juggle project management tools, file sharing, compliance apps, and more, the ability to spread tasks across multiple processors isn't just helpful—it's non-negotiable. A modern general-purpose OS brings real power to the table through symmetric multiprocessing, giving the system the brains to allocate threads and processes dynamically across all available CPU cores. No more idle processors waiting around while one core does all the heavy lifting—the scheduler knows what's running, what's waiting, and how to balance the load based on priority and availability. That means smoother multitasking, fewer slowdowns, and better overall responsiveness. For TSI, this isn't just about performance—it's about meeting client expectations, staying competitive, and operating with the same precision internally that they demand from the solutions they deliver. Upgrading to an OS that understands how to command a multiprocessor setup gives TSI the confidence that every core is working, every cycle counts, and no task is left behind. "The concept of SMP is applied to multi-core processors, where each core is treated as a separate processor (Symmetric multiprocessing 2025)."

Memory Abstraction

TSI understands that a general-purpose operating system must manage memory intelligently to support its complex, multitasking back-office environment. The OS supports multiple types of memory, including physical RAM, virtual memory, cache, and swap space, each working together to optimize performance and stability. Virtual memory is the core abstraction, allowing applications to operate as if they have access to a large, continuous block of memory, regardless of physical RAM limitations. This abstraction isolates processes, enhances security, and prevents system crashes due to memory exhaustion. Using tools like vmstat (a command-line tool on Unix-like systems that reports on virtual memory statistics, including processes, memory, paging, I/O, and CPU activity) (Carrigan, 2020) or Windows Resource Monitor "Resmon is a built-in Windows tool that allows you to monitor and analyze the resource usage of your computer. It provides detailed information about the central processing unit (CPU), memory, disk, and network performance of your system. (What is a resource monitor 2025)", TSI can analyze how virtual addresses are mapped to physical memory, revealing how efficiently memory is being used and whether resources are being overcommitted. These insights help the team fine-tune application behavior, reduce memory leaks, and ensure that the system performs reliably—even under peak load.

Access Activity

TSI recognizes that effective memory management is critical for sustaining performance under heavy workloads, especially as its operations scale. A general-purpose operating system offers robust support for virtual memory, paging, and segmentation, each with distinct advantages. Memory management is a critical challenge for the operating system. It allows for properly using a computer's memory resources, including the system, programs, and data storage. If memory is not properly managed, TSI's system may have difficulty handling more than one or two applications. Memory management generally enables a computer to perform several tasks simultaneously by dividing memory areas among processes. (Memory management in OS) Disadvantages of memory management technologies include performance overhead from processes like swapping and garbage collection, fragmentation of memory, and security vulnerabilities like buffer overflows from incorrect implementation. Additionally, some methods can be overly complex, inefficient for certain workloads, or have limitations, such as static memory's inability to adapt to changing needs. (PaulPaul, 2014) Virtual memory allows applications to exceed physical RAM limits, improving multitasking and stability, while paging breaks memory into manageable blocks that the OS can load or unload as needed, optimizing physical memory use. Segmentation further organizes memory by separating code, data, and stack, enhancing security and flexibility. However, excessive paging can lead to performance degradation, known as thrashing, and segmentation complexity may introduce overhead. Using monitoring tools like Windows Performance Monitor, TSI can observe page faults, swap activity, and segment usage in real time to evaluate how well the system handles memory pressure. These insights allow TSI to fine-tune configurations and ensure the OS maintains responsiveness and efficiency during peak operational demand.

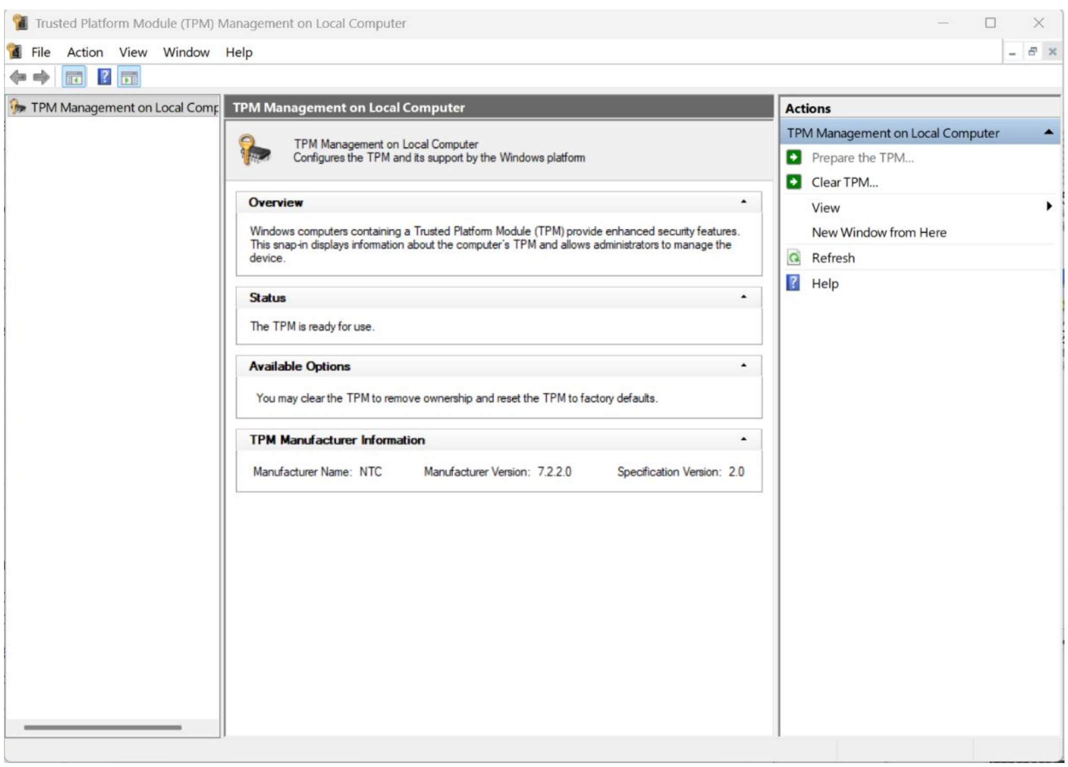
Techniques

TSI values the separation of memory management policy and mechanism as a critical design principle in modern general-purpose operating systems, enabling greater control and reduced complexity in system behavior. The mechanism—how memory is allocated, paged, or swapped—is built into the OS kernel of Windows and operates at a low level for efficiency, while the policy—decisions about when and how to allocate or reclaim memory—is more flexible and can be tuned based on system needs. This separation allows TSI to adjust memory usage strategies without altering core kernel functions, making it easier to respond to workload changes or performance issues. By keeping policy and mechanism decoupled, the system remains modular, easier to manage, and adaptable to the dynamic demands of TSI’s growing operations.

Hardware/Software Interface

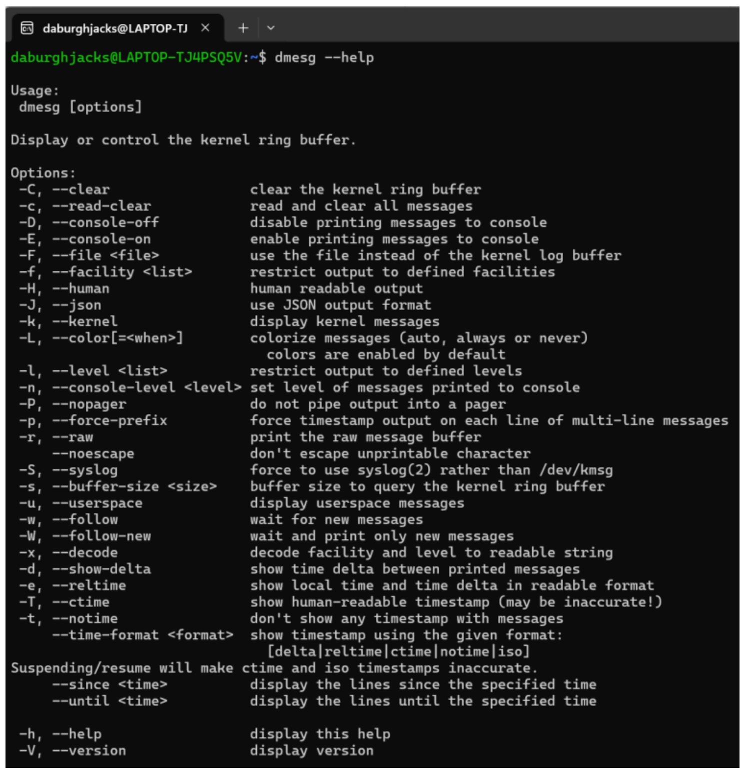
Once implemented onto TSI system the hardware-software interface of a general-purpose operating system to manage communication with diverse office devices efficiently and reliably would adhere to business operations. The OS supports several I/O techniques:

programmed I/O, interrupt-driven I/O, and direct memory access (DMA). Programmed I/O is straightforward and easy to implement, but it keeps the CPU occupied while waiting for I/O operations to complete, making it inefficient for multitasking environments like TSI's. Interrupt-driven I/O is more efficient, allowing the CPU to process other tasks until the device signals completion; however, excessive interrupts can cause processing delays and increased context-switching overhead. DMA provides the highest performance by enabling devices to transfer data directly to memory without burdening the CPU, though it adds hardware complexity and may be overkill for simpler tasks. Each method is supported by the OS through device drivers, which abstract hardware control and allow TSI to plug in devices without writing custom code. Using Windows Event Viewer to view and analyze system logs for troubleshooting, security monitoring, and performance analysis;



+R then Enter.

Or tools like `iostat`- a command-line utility that monitors system input/output (I/O) device loading by reporting CPU and disk I/O statistics. It is used by system administrators to analyze performance, identify bottlenecks, and balance I/O loads between physical disks., or `dmesg`- command in Linux is used to display and control the kernel ring buffer, which stores messages generated by the kernel during system boot and runtime. These messages provide crucial information about hardware detection, device initialization, and any potential issues;

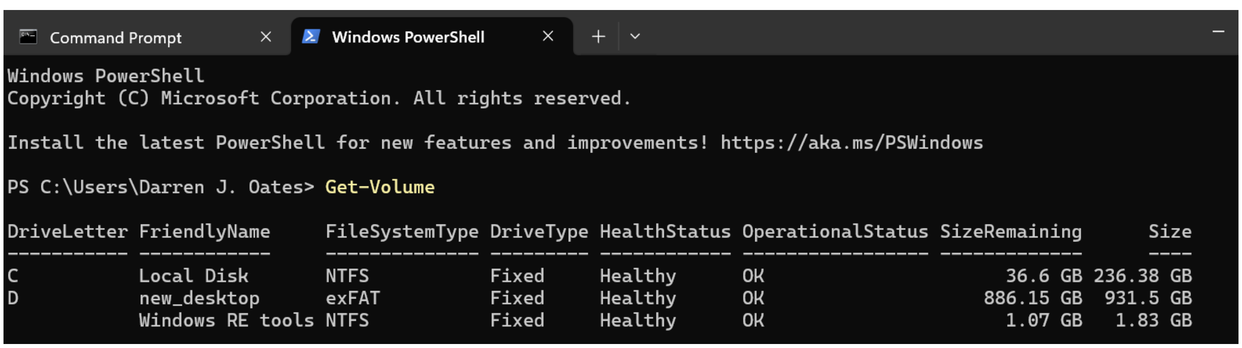


```
daburghjacks@LAPTOP-TJ4PSQ5V:~$ dmesg --help

Usage:
 dmesg [options]

Display or control the kernel ring buffer.

Options:
 -C, --clear                 clear the kernel ring buffer
 -c, --read-clear            read and clear all messages
 -D, --console-off           disable printing messages to console
 -E, --console-on            enable printing messages to console
 -F, --file <file>           use the file instead of the kernel log buffer
 -f, --facility <list>       restrict output to defined facilities
 -H, --human                 human readable output
 -J, --json                  use JSON output format
 -k, --kernel                display kernel messages
 -L, --color[=<when>]        colorize messages (auto, always or never)
                               colors are enabled by default
 -l, --level <list>          restrict output to defined levels
 -n, --console-level <level> set level of messages printed to console
 -P, --nopager               do not pipe output into a pager
 -p, --force-prefix          force timestamp output on each line of multi-line messages
 -r, --raw                   print the raw message buffer
     --noescape              don't escape unprintable character
 -S, --syslog                force to use syslog(2) rather than /dev/kmsg
 -s, --buffer-size <size>    buffer size to query the kernel ring buffer
 -u, --userspace             display userspace messages
 -w, --follow                wait for new messages
 -W, --follow-new            wait and print only new messages
 -x, --decode                decode facility and level to readable string
 -d, --show-delta            show time delta between printed messages
 -e, --reltime               show local time and time delta in readable format
 -T, --ctime                 show human-readable timestamp (may be inaccurate!)
 -t, --notime                don't show any timestamp with messages
     --time-format <format>  show timestamp using the given format:
                               [delta|reltime|ctime|notime|iso]
Suspending/resume will make ctime and iso timestamps inaccurate.
     --since <time>          display the lines since the specified time
     --until <time>          display the lines until the specified time

 -h, --help                  display this help
 -V, --version               display version
```

Using these tools, TSI can monitor I/O activity and troubleshoot driver issues, helping balance performance, reliability, and complexity across its infrastructure.

File Systems

TSI recognizes that Windows operating system utilizes the New Technology File System (NTFS), a robust and secure file system designed to support large volumes, advanced permissions, and modern storage demands. NTFS provides key features such as file-level encryption, compression, disk quotas, and journaling, which help prevent data corruption and ensure system reliability after crashes or power failures. It uses a Master File Table (MFT)* to store metadata, allowing quick access and efficient indexing of files, which is essential for TSI's data-intensive back-office operations. NTFS also supports Access Control Lists (ACLs) for granular security management, aligning with TSI's compliance needs when handling sensitive client information. While NTFS offers superior security and stability compared to older systems like FAT32, it can be less efficient for smaller removable drives or cross-platform compatibility. Overall, Windows 11's NTFS file system provides TSI with the scalability, resilience, and security necessary to manage growing datasets while maintaining the high performance required for daily business operations. 'Get-Volume' command line will display the file types.



Context Switching and I/O Interrupt Handling

Scheduled process context switching and I/O interrupt handling are closely linked functions within a modern general-purpose operating system, working together to maintain responsiveness and efficiency. When an I/O interrupt occurs—such as a device signaling that it has completed a task—the operating system temporarily suspends the running process and performs a context switch to handle the interrupt through an interrupt service routine. This mechanism allows the CPU to respond immediately to hardware events without waiting for ongoing processes to finish, improving overall system performance. Compared to programmed I/O, where the CPU waits idly for device operations to complete, interrupt-driven I/O allows better multitasking and resource utilization. Direct Memory Access (DMA) further reduces CPU overhead by transferring data directly between devices and memory, though it adds hardware complexity. For TSI, the use of interrupt-driven I/O combined with efficient context switching ensures that mission-critical applications remain responsive under heavy workloads while balancing CPU efficiency and system throughput.

Security Model

Operating system functions that must be part of the TCB (Trusted Computing Base) include process creation, process switching, memory management, and part of file and I/O management. Since TSI controls and sells its operating system to Fortune 500 companies and government agencies it should use hardware and software that is trusted. The structure of the operating system has important consequences for the security guarantees. In Windows 11 (and Linux), The TCB consists of all the code running in the kernel. (Tanenbaum, A. S, 2022)

Recommended Techniques

TSI should use the Unikernels model. Here, a minimal kernel is responsible only for partitioning the resources at the lowest level, but all operating system functionality required for the single application is implemented in the application's security domain in the form of a minimal ''LibOS.'' The design allows applications to customize the operating system functionality exactly according to their needs and leave out everything that they do not need anyway. Doing so reduces the attack surface. While you may object that running everything in the same security domain is bad for security, do not forget that there is only a single application in that domain—any compromise will affect only that application. (Tanenbaum, A. S, 2022)

Overall Evaluation

Based on the defined requirements, analyses, and assessments, it is recommended that TSI adopt a modern general-purpose operating system (GPOS), such as **Windows 11**, to meet its operational, technical, and security needs. This OS provides comprehensive support for **multiprogramming, multiprocessing, multithreading, and virtual memory**, enabling efficient utilization of CPU and memory resources even on legacy or upgraded hardware. Its **preemptive multitasking and advanced scheduling** ensure responsiveness during peak workloads, while **NTFS file systems** and integrated **security features**—including access control lists, encryption, and auditing—safeguard sensitive government and enterprise data. Windows 11's **robust driver and I/O management** ensures compatibility with TSI's SATA and SCSI storage systems, while tools such as **Windows Task Manager**, **Resource Monitor**, and **Windows Performance Analyzer** allow continuous performance monitoring and optimization. With support for **fault-tolerant storage (RAID, snapshots, failover)** and modular system architecture, the OS offers scalability, reliability, and simplified maintenance. Overall, Windows 11's architecture and toolset align with TSI's organizational goals—enhancing system stability, developer productivity, and security—while ensuring the infrastructure can support future growth and enterprise-grade performance.

References

Alvinashcraft. (n.d.). *Master file table* . Win32 apps | Microsoft Learn. https://learn.microsoft.com/en-us/windows/win32/fileio/master-file-table

Carrigan, T. (2020b, June 10). Linux commands: Exploring virtual memory with vmstat. Red Hat - We make open source technologies for the enterprise. https://www.redhat.com/en/blog/linux-commands-vmstat#:~:text=Virtual%20memory%20statistics%20reporter%2C%20also,the%20array%20of%20information%20provided.

Documentation. VisualVM. (n.d.). https://visualvm.github.io/documentation.html

Memory management in OS: Importance and techniques. Hero Vired. (n.d.-b). https://herovired.com/learning-hub/blogs/memory-management-in-os/

PaulPaul. (2014, March 9). *Disadvantages of scoped-based memory management*. Software Engineering Stack Exchange. https://softwareengineering.stackexchange.com/questions/231789/disadvantages-of-scoped-based-memory-management

Tanenbaum, A. S., & Bos, H. (2022). Modern Operating Systems (5th ed.). Pearson Education (US). https://mbsdirect.vitalsource.com/books/9780137618941

What is a resource monitor (resmon) and how does it work? | lenovo us. (n.d.). https://www.lenovo.com/us/en/glossary/resource-monitor/

Wikimedia Foundation. (2025, August 13). *Symmetric multiprocessing.* Wikipedia. https://en.wikipedia.org/wiki/Symmetric_multiprocessing#:~:text=Symmetric%20m ultiprocessing%20or%20shared%2Dmemory,treating%20them%20as%20separate%20processors.

Windows-Driver-Content. (n.d.). *Windows performance analyzer.* Microsoft Learn. https://learn.microsoft.com/en-us/windows-hardware/test/wpt/windows-performance-analyzer

* The NTFS file system contains a file called the master file table, or MFT. There is at least one entry in the MFT for every file on an NTFS file system volume, including the MFT itself. All information about a file, including its size, time and date stamps, permissions, and data content, is stored either in MFT entries, or in space outside the MFT that is described by MFT entries. (Alvinashcraft, 2024)