Comparison of The Two Most Popular Operating Systems: Microsoft Windows & Apple macOS

Completed Date: 13th December 2024

Table of Contents

Summary

This review compares, contrasts and critically evaluates two modern desktop operating systems: Microsoft Windows and Apple MacOS. Focused on describing and evaluating their implementation of threading, memory management and security within their systems combined with identifying their potential benefits and pitfalls. A simplified understanding should be generated on how each modern operating system use various techniques and specialized features relevant to implementing the components.

Windows 11 and macOS Sonoma are discussed as the representatives of the OSes.

CHAPTER ONE: Introduction

Modern operating systems have significantly contributed to the computing world, providing useful communication between humans and digital computer systems along with reliable management of software and hardware processes. The book "Operating System Concepts 10th Edition" states in the preface that the computing field is "undergoing rapid change, as computers are now prevalent in virtually every arena of day-to-day life" which refers to modern operating systems and it could not be truer (Silberschatz, Galvin and Gagne, 2018). Figure 1 presents a bar chart displaying the amount of computer households worldwide from 2014 to 2029 with an estimated number around 996.42 million households with a computer in 2023. This statistic is expected to increase up to 1.1 billion in 2029 (Statista, 2024). It is common for an OS to be pre-loaded within a personal computer therefore the hypothesis can be generated to state that the usage of modern operating systems will continue to rise in the future.

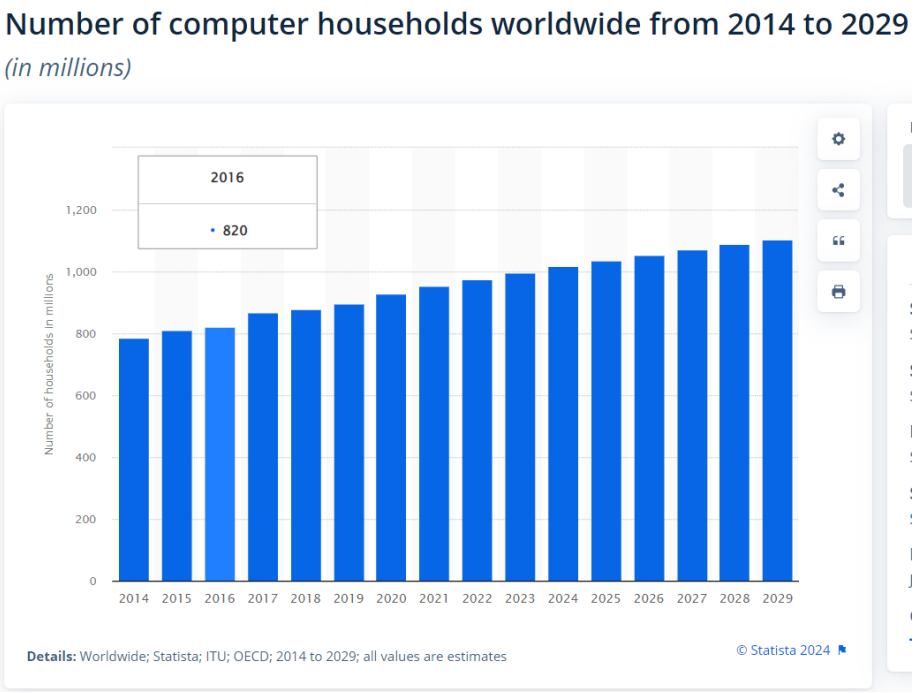

Figure 1 – number of computer households worldwide from 2014 to 2029 (Statista)

1.1. Overview of the Two Operating Systems

The two modern operating systems which will be discussed in this review:

- Microsoft Windows (Windows 11)

- Apple MacOS (Sonoma)

Windows 11 is the latest Windows NT operating system version developed by Microsoft and released on October 5th 2021, maintaining and improving previous Window features on security, graphics and general user experience with enhanced performance due to hardware improvements along with better energy efficiency for longer battery life (Windows, 2024). Similarly, macOS Sonoma shines as the latest version of the macOS developed by Apple and released on September 26th 2023, improving user experience, performance and security such as secure password sharing and optimized graphical processor power for gaming (Mac App Store, 2024). Both modern operating systems selected operate in the same device for an accurate comparison and evaluation.

(All OSes reviewed are the latest versions during the creation of the literature review)

1.2. Scope of the Review

The review explores several crucial components vital for all modern operating systems to function from processes to memory management however three will be specifically evaluated on the Microsoft Windows and Apple macOS operating systems, highlighting benefits and drawbacks based on how they implement these components. Threading, memory management and security are the primary factors identified due to their common significant involvement in maintaining high-quality performance during data processing and management operations within these systems. Windows 11 and macOS Sonoma are strictly dwelled on deeply however any general characteristics and features commonly derived from the main operating system is briefly mentioned. It is clear that any reference to Windows 11 refers to the standard edition of Windows 11 Home for personal use. Finally, academic or relevant sources referred are recently released and limited due to lack of general research on the latest operating systems.

CHAPTER TWO: Literature Review – Comparison & Analysis of the Two Operating Systems

2.1. Threading

Many relevant vital concepts and techniques should be thoroughly understood like processes to reveal their relations with threads and their importance. Every operating

system uses processes to commit essential tasks and manage system resources for software programs, files, etc. They can be described as constant “workstations” containing a thread – defined as a mini-process which executes source code rapidly to allow multiple tasks within a process to run on the processor simultaneously, offering parallel processing. Threading is a key process within processes yet it is completely independent from it, sharing resources and memory space with each other while performing concurrency for utilized performance as processor response times reduce with multiprocessing with additional concepts (*Tanenbaum and Bos*, 2015).

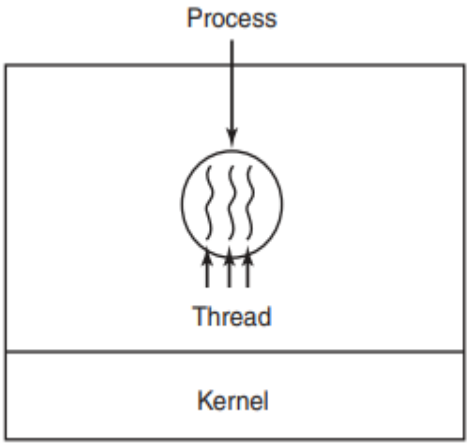

Figure 2 – Multithreading Diagram (Tanenbaum & Bos, 2015)

Multithreading & Thread Management

Most modern operating systems are compatible with multithreading shown on Figure 2 – a technique involving multiple threads to execute tasks in the same process, utilizing multi-core processors (Tanenbaum & Bos, 2015). Windows 11 and Sonoma implement multithreading in a similar manner however they differ on thread management. Like Windows 10 beforehand, the latest OS version embeds the Window NT kernel which directly manages threads without any automatic thread management code rather using an application programming interface named Win32. Several functions included in the API such as “GetThreadContext” and “CreateThread” assist the operating system with casual threading. Win32 provides freedom for advanced developers to manually manage crucial OS components like threading and file management however this acts as a double-edged sword as it raises general OS management and security concerns, increasing the chances of various vulnerabilities like buffer overflow when managed poorly (Yosifovich et al, 2017). On the other hand, Sonoma is restrictive compared to Windows 11 which developers have limited control over Grand Central Dispatch - an API that specializes specifically with multithreading on macOS rather than permitting complete OS management. Instead of directing threads firsthand, it involves adding

tasks to dispatch queues to be executed concurrently like “dispatch_async” by reused groups of worker threads called a thread pool (Apple Developer Documentation, 2023).

Thread Scheduling

Another characteristic involves thread scheduling. Both operating systems employ pre-emptive scheduling by using a priority-based system with the round robin algorithm. Windows 11 uses priority level system to schedule threads with higher priority to execute significant tasks first while establishing equality between threads, assigning time using a round-robin algorithm. Typically, each thread is labelled with a priority level numbered in the range of 0 to 31 which 0 is the lowest and 31 is the highest (Karl-Bridge-Microsoft, 2021). Similar to Windows 11, Sonoma implements a priority-level system to dictate thread priority using Quality of Services levels to class threads in categories from real-time to background for better energy efficiency (Apple, 2016).

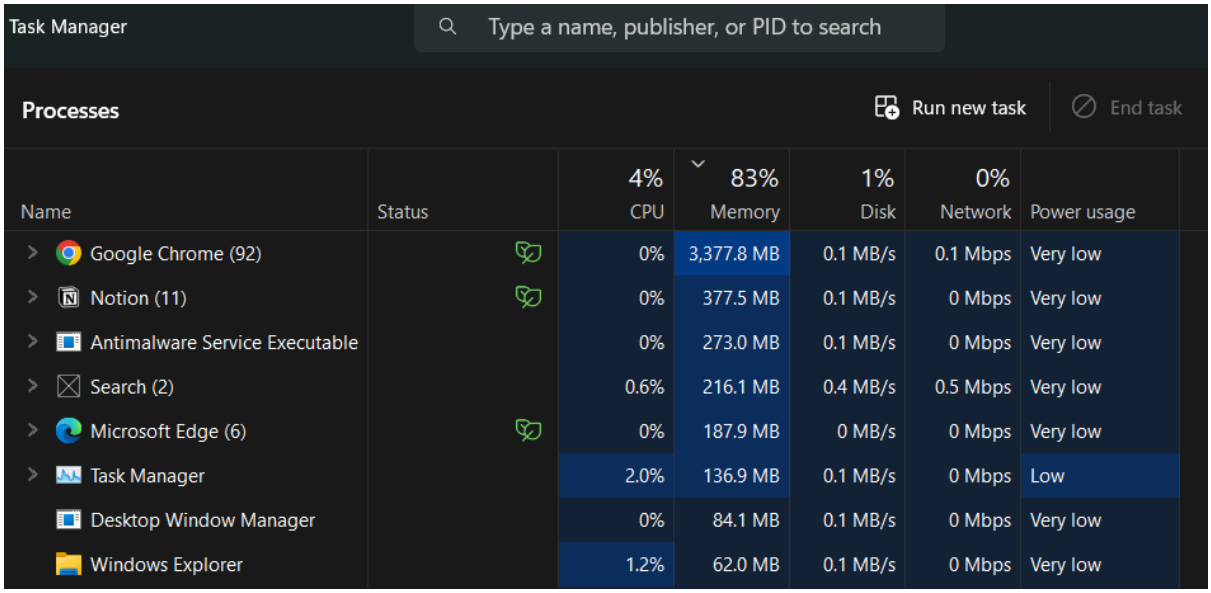

Figure 3 – Windows 11 Task Manager with processes and characteristics

Figure 4 – Windows Anti-Malware Service Process and Thread

Windows 11 and Sonoma successfully configures threading and key relevant techniques well yet indirectly reveal their intentions on usage and structure. Windows 11 seems attractive yet complex for developers, revealing a steep learning curve to understand threading configuration but follows the traditional formula to allow manual liberty to manage general OS management. In Figure 3 and 4, users can open Windows Task Manager to view the processor utilization, memory, network and power consumption of running real-time processes and their threads with the option to end them. Sonoma focuses on maintaining their system without external modifications while formulating and implementing an innovative progressive model on threading.

2.2. Memory Management

Memory management formulates the basis of maintaining any system. Windows 11 and macOS Sonoma show no difference in defying this component which is essential for resource allocation in main memory (RAM). Many concepts fall under memory management from applying paging to virtual memory which are significant to consider as these systems allocate and deallocate memory space to a large number of processes. For example, a typical Windows or macOS computer execute around 50 – 100 processes with each process possessing around 5 - 10MB as it boots up, implying a vast amount of memory space is allocated initially (*Tanenbaum and Bos*, 2015).

Paging, Segmentation & Virtual Memory

Both operating applies memory management significantly similar. Windows 11 and Sonoma introduces dynamic partitioning which each process is loaded into a partition which matches the memory size of the process. This improves memory efficiency with no internal fragmentation issues. It is usually caused when a process is allocated more memory than required however processor utilization reduces from compaction. Figure 5 shows memory organised in a macOS device, presenting a default system partition and the creation of application partitions with their own stack and heap portions as a process executes due to launched applications on the device (Apple, 2023).

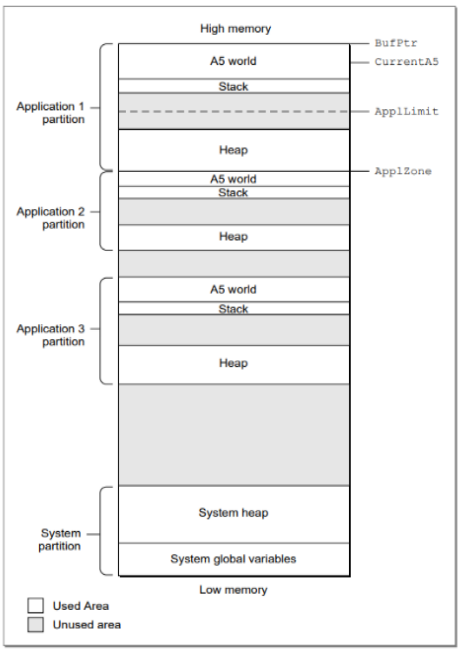

Figure 5 – the memory organisation of opened applications in a macOS device

Simple paging involves dividing main memory into equal frames. Processes are divided equally into pages in a page file. Once a process is required, the pages of that process will be loaded. This is useful for managing virtual memory, helping allocate space for real-time processes beyond the memory space capacity of physical main memory. Every Windows OS version, including Windows 11 have a default system page file – the "sys" file for paging on disk while using Virtual Memory Manager to provide virtual address space for processes using the concept of virtual memory paging (Yosifovich et al, 2017). Additionally, segmentation is applied with paging on Windows 11 however it acts as a secondary technique. Sonoma relies on compaction of memory space before paging for virtual memory with no aspect of using segmentation but this results in a reduction of processor utilization (Apple, 2023). Both OSes receive the benefit against a high chance of experiencing external fragmentation along with efficient memory management for space but overhead issues may occur due to complexity issues.

2.3. Security

Jaeger (2022) states that the challenges of implementing security within operating systems includes protecting process execution and the relevant data during complex interactions using security mechanisms that will guarantee the adherence of security

requirements. He labels the concept of a secure operating system as an “ideal and an oxymoron" explaining that an operating system cannot become completely secure due to development errors in the system and the difficulty of tracking those errors/bugs. The mechanisms must ensure reliable protection of data from unauthorized access by following the C.I.A triad which provides key considerations when introducing security.

Process Isolation & Resource Management

It is vital that security measures in the operating system must enable confidentiality, availability and integrity during system operations. Windows 11 and Sonoma offer process isolation and resource management techniques by applying authorization and authentication. Both Oses use several features for process isolation to provide internal security for the kernel like Hypervisor-Protected Code Integrity and Virtualisation-Based Security on Windows while macOS includes System Integrity Protection on macOS. It ensures protection to isolate and restrict low priority processes from crucial system files, avoiding system tampering and exploits while offering integrity and secure authorization (Rathbone, 2021) (Apple, 2024). Windows labels a unique security identifier (SID) to a user with applicable processes and threads for validation on every system boot (Silberschatz, Galvin and Gagne, 2018). Similarly, macOS enables memory tagging extensions (MTE) on processes, assisting with quick debugging for developers.

Security Tools

Windows has been infamous for being a constant target for cyberattacks due to being popular along with permitting OS control to developers (mentioned previously in Threading) therefore it has improved their system security configuration over the years. This is seen through Trusted Module Platform 2.0 which is a security feature which protects Windows 11 hardware by involving encryption using effective cryptography algorithms. Windows Defender is used to frequently prevent malicious software and networking threats during runtime (Yosifovich et al, 2017). Sonoma has similar features to TMP and Windows Defender such as FileVault and XProtect. They provide the same functionalities which TMP and Microsoft Defender offers by encrypting the storage disk, monitoring network traffic and issuing background authentication during runtime (Apple, 2022). Sonoma limits third-party involvement which maintains strong privacy.

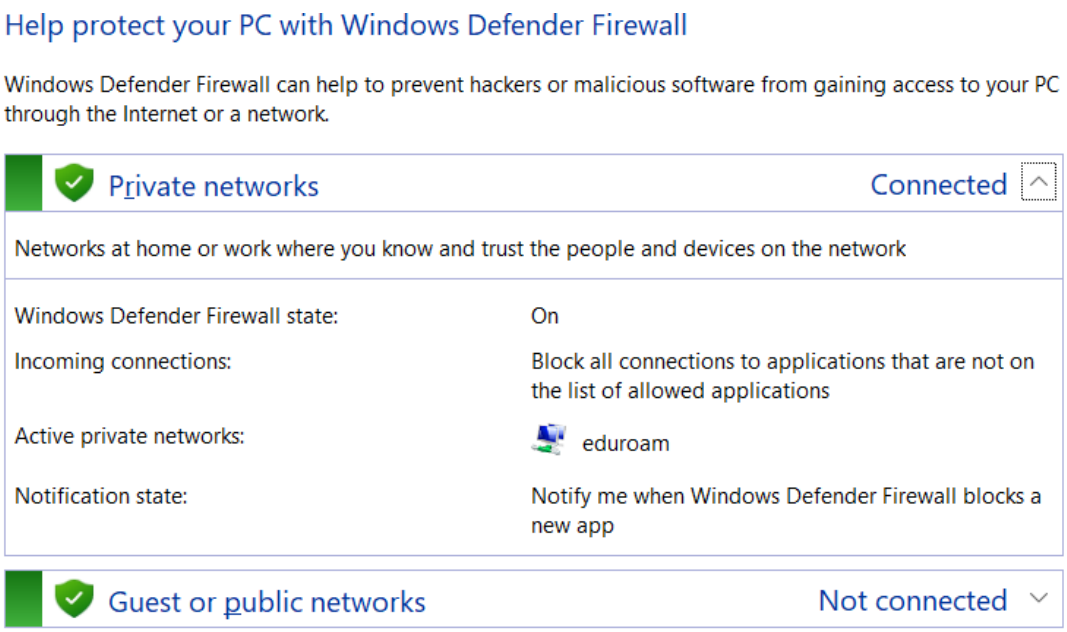

Figure 6 – Windows Defender Firewall

Overall, security is evidently implemented into these systems to prevent and mitigate several system vulnerabilities and malicious threats. Windows 11 prioritises on running complex security measures on hardware and applying constant authentication and authorization. To differ, Sonoma's approach is simplistic compared to Windows 11, applying security to provide good user experience and isolate their hardware.

CHAPTER THREE: Conclusion

At the end, all three components are successfully configured in Windows 11 and macOS Sonoma with similar techniques on implementation but reveal underlying differences on features, their intentions and consequences. The Windows operating system grants freedom and promotes creativity by permitting users to indulge in monitoring these components, whether they are refined or neglected by third-party involvement. They formulate a reliable optimized architecture of multiple features and techniques on threading, memory management and security initially. This is why Windows is one of the most popular efficient desktop operating systems around for personal use and enterprise. However, this relaxed approach of manual management of a complicated operating system leads to long-term security concerns as new technologies could be introduced to adapt against updating Window security features. MacOS shows innovation and self-reliance to develop unique techniques and subsystems to implement these components. Numerous benefits such as security for users and high-quality performance of specialized hardware has labelled the macOS operating system as the “aesthetic operating system”. However, the OS lacks customization compared to Windows and sticks limitations on potential enhancements of the system by failing to permit certain hardware and applications.

These two modern operating systems will continue to innovate new effective features and dominate the computing field for a prolonged amount of time until a mysterious concept is developed and integrated into a new OS which may outperform the two.

Reference List

- Silberschatz, A., Galvin, P.B. and Gagne, G. (2018). Operating system concepts. Hoboken, Nj: Wiley. Accessed: September 30, 2024

- Statista. (2024). *Number of computer households worldwide from 2014 to 2029 (in millions)*. *Statista*. Statista Inc. Accessed: September 30, 2024. https://www.statista.com/forecasts/1146269/computer-households-in-the-world

- Windows, M. (2024). *Meet Windows 11 - Microsoft Support*. [online] Microsoft.com. Accessed: September 30, 2024 https://support.microsoft.com/en-gb/meetwindows11

- *Mac App Store. (2024). macOS Sonoma. [online] Accessed: September 30, 2024 https://apps.apple.com/us/app/macos-sonoma/id6450717509?mt=12*.

- *Tanenbaum, A.S. and Bos, H. (2015). Modern Operating Systems, Global Edition. Pearson Higher Ed. Accessed: November 15, 2024*

- Yosifovich, P., Solomon, D.A., Russinovich, M.E. and Ionescu, A., 2017. *Windows Internals: System architecture, processes, threads, memory management, and more, Part 1*. Microsoft Press. Accessed: November 23, 2024

- *Apple Developer Documentation*. (2023) | *Apple Developer Documentation*. [online] Accessed: December 2, 2024 https://developer.apple.com/documentation/DISPATCH

- Karl-Bridge-Microsoft (2021). *Scheduling Priorities - Win32 apps*. [online] learn.microsoft.com. Accessed: December 10, 2024 https://learn.microsoft.com/en-us/windows/win32/procthread/scheduling-priorities.

- Apple.com. (2016). *Energy Efficiency Guide for Mac Apps: Prioritize Work at the Task Level*. [online] Accessed: December 10, 2024 https://developer.apple.com/library/archive/documentation/Performance/Conceptual/power_efficiency_guidelines_osx/PrioritizeWorkAtTheTaskLevel.html

- Apple.com. (2023). Introduction to Memory Management. [online] Accessed: December 11, 2024 https://developer.apple.com/library/archive/documentation/mac/pdf/Memory/Intro_to_Mem_Mgmt.pdf

- *Jaeger, T. (2022). Operating System Security. Springer Nature. Accessed: December 11, 2024*

- *Rathbone, A. (2021). Windows 11 For Dummies. John Wiley & Sons. Accessed: December 12, 2024*

- Apple Developer Documentation. (2024). *Disabling and Enabling System Integrity Protection | Apple Developer Documentation*. [online] Accessed: December 12, 2024 https://developer.apple.com/documentation/security/disabling-and-enabling-system-integrity-protection.

- Apple (2022). *Apple Platform Security*. [online] Accessed: December 13, 2024 https://help.apple.com/pdf/security/en_US/apple-platform-security-guide.pdf.