ISSN: 2395-7852

International Journal of Advanced Research in Arts, Science, Engineering & Management

Volume 12, Issue 1, January- February 2025

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.583

+91 9940572462 +91 9940572462 ijarasem@gmail.com www.ijarasem.com

International Journal of Advanced Research in Arts, Science, Engineering & Management (IJARASEM)

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 7.583 | Bimonthly, Peer Reviewed & Referred Journal|

| Volume 12, Issue 1, January- February 2025 |

Comparative Analysis of Operating System Fundamentals: A Review of Core Concepts and Functionalities

Realuz S. Aliguay, Jerry I. Teleron

0009-0003-4325-8765, 0000-0001-7406-1357

Department of Graduates Studies, Surigao Del Norte State University, Surigao City, Philippines

ABSTRACT: This study focuses on comparative analysis of the fundamental concepts and functionalities of modern operating systems (OS). Operating systems (OS) form the backbone of modern computing, acting as intermediaries between hardware and software, managing resources and providing essential services. The complexity and variety of operating systems available today necessitate a thorough understanding of their core concepts and functionalities. This paper presents a comparative analysis of various operating system fundamentals, exploring core components such as process management, memory management, file systems, input/output operations, and security mechanisms. Through a detailed examination of different OS families (Windows, Linux, macOS, and Unix-based systems), this review highlights the strengths, weaknesses, and use-case scenarios of each, offering a comprehensive framework for evaluating operating system performance and suitability for diverse computing environments.

KEYWORDS: Operating System, Comparative Analysis, Memory Management, Process Management, Operating Design System.

I. INTRODUCTION

Operating systems (OS) are critical software components that manage computer hardware and software resources, providing a seamless interface for users and applications to interact with computing systems. They play an essential role in ensuring efficient and reliable system operations by handling tasks such as memory management, process scheduling, and input/output operations Idris et al.,[1]. As computing technology advances, a deeper understanding of OS fundamentals has become indispensable for developers, system administrators, and IT professionals to optimize system performance and usability.

The study of operating system fundamentals encompasses key concepts such as process management, memory management, file systems, security, and user interfaces. Emerging trends include IoT operating systems, cloud-based OS, AI-driven OS, blockchain-based OS, hybrid OS, and containerized OS. These advancements reflect how operating systems adapt to evolving technological landscapes Bazukiu et al.,[2]. Each OS employs unique methods and algorithms to manage resources, resulting in significant differences in performance, scalability, and usability across platforms such as Windows, Linux, macOS, Android, and iOS.

The growing demands of cloud computing, virtualization, and mobile computing have led to the development of new OS technologies and methodologies. Additionally, the integration of cutting-edge hardware architectures, such as multi-core processors and IoT devices, has introduced complexities that challenge traditional OS models Zhang & Chen,[3]. These developments require a reevaluation of core operating system principles to ensure relevance in addressing modern computing challenges.

A comparative analysis of operating systems is crucial for highlighting similarities and differences among platforms, providing insights into how various OS types address common challenges. Key areas such as task scheduling, resource allocation, and security protocols are examined to understand how operating systems evolve and adapt to meet the needs of an ever-changing technological environment Ahmed, S. & Patel, R. [4].

II. LITERATURE REVIEW

Research on operating systems (OS) has been extensive, covering fundamental concepts, design philosophies, and their practical applications across various computing environments. This section reviews significant studies that contribute to the understanding of OS fundamentals, including process management, memory management, file systems, and security models, while highlighting emerging trends and comparative analyses.

Process Management and Scheduling

Recent studies have explored process management and scheduling as core functionalities of operating systems. Adekotujo, M. et al. [5] conducted a comparative analysis of operating systems, providing insights into their features and strengths to guide both developers and end-users in selecting suitable platforms. Furthermore, specialized operating systems tailored for supercomputers have been highlighted for their role in optimizing high-performance computing environments Accamma, C.G., et al.,[6].

Chien, C.F. et al. [7] [8] emphasized the importance of user experience (UX) in OS design, proposing a framework based on real-world needs. Their empirical study using popular tablet OSs identified critical factors influencing user satisfaction, offering practical design guidelines for improving user-centric operating systems. Xiong et al. [9] examined the evolution of cloud operating systems for industrial applications, presenting a framework for integrating IoT, cloud computing, and big data into intelligent industrial systems.

Gaur, P., and Tahiliani, M., [10] focused on operating systems designed for IoT devices, proposing a generic framework that highlights essential features like resource efficiency and scalability. Their findings underscore the limitations of traditional OS platforms like Windows or Unix in addressing IoT requirements and advocate for lightweight alternatives such as Contiki and FreeRTOS.

Memory Management and File Systems

Memory management continues to be a critical focus area in OS research. Omar, N.R., et al. [11] explored techniques for managing memory and reclaiming space within operating systems, emphasizing the role of the memory management unit in improving system efficiency. Similarly, Srinuan, P., et al. [12] introduced Cooperative Memory Expansion (COMEX), an OS kernel extension that creates a shared memory pool across nodes, enhancing scalability and addressing resource imbalances in data centers.

Musaddiq, A et al. [13] analyzed lightweight IoT operating systems and their strategies for managing constrained resources, such as memory and energy. Their review of Contiki, TinyOS, and FreeRTOS highlights these OSs' strengths and limitations in managing IoT-specific workloads.

In the realm of file systems, Gu, C. et al. [14] identified vulnerabilities in isolation technologies like containers and processes, demonstrating file system side-channel attacks that compromise logical isolation. This research underscores the critical role of file systems in managing I/O operations while maintaining data security. Rodeh, O., et al. [15][16] examined the Btrfs file system, showcasing its adaptability across diverse workloads through efficient snapshots, cloning, and sustained performance.

Security and Access Control

Security remains a cornerstone of OS research. Mohamed, A.K.Y.S et al. [17] reviewed advanced access control models for safeguarding sensitive information against unauthorized access, while Zarif, K. et al. [18] emphasized the importance of OS security in maintaining data confidentiality and integrity. Wenrui, D. et al. [19] revealed novel file system side channels that compromise OS isolation mechanisms, presenting practical solutions to mitigate such vulnerabilities.
Park, H. [20] introduced AvaTar, a novel file archiving system leveraging zero-copy merging and splitting operations at the kernel level. This system significantly improved file extraction and cloud upload speeds, demonstrating advancements in kernel-level resource management for enhanced security and efficiency.

Comparative Studies and Emerging Trends

Bazukiu et al. [2] analyzed emerging OS trends, including IoT, cloud, AI-driven, blockchain, hybrid, and container operating systems. Their study proposed the concept of a universal OS adaptable to all architectures, incorporating green computing principles to address power consumption challenges [21].

Thangavel, R. [22] [23] provided a comparative survey of Windows, Linux, and macOS, focusing on memory management, architecture, security, and versatility. By examining their core features, the study offered insights into the unique strengths and weaknesses of these systems, highlighting their relevance in different computing environments.

Synthesis

Research on operating systems has evolved significantly, addressing core functionalities like process management, memory management, file systems, and security while adapting to emerging trends in IoT, cloud computing, and big data. Comparative studies and innovative frameworks have expanded the understanding of OS design and applications, offering valuable insights for both academic and industrial domains. These advancements underscore the importance of ongoing research in optimizing operating systems to meet the demands of modern computing environments.

III. OBJECTIVES OF THE STUDY

The study aims to achieve the following:

1. To compare the core concepts of operating systems, including process management, memory management, file systems, and input/output operations, across various popular operating systems.
2. To evaluate the functional differences and similarities in how operating systems handle key tasks such as scheduling, resource allocation, and system security.
3. To assess the impact of different operating system architectures on system performance, scalability, and efficiency in various computing environments.
4. To explore the evolution of operating system fundamentals and how advancements in technology have influenced the development of modern operating systems.

IV. METHODS

This study employs a comparative analysis methodology to evaluate the core concepts and functionalities of different operating systems (OS). The focus is on identifying key similarities and differences between OS principles, architecture, system performance, user interface design, resource management, and security mechanisms.

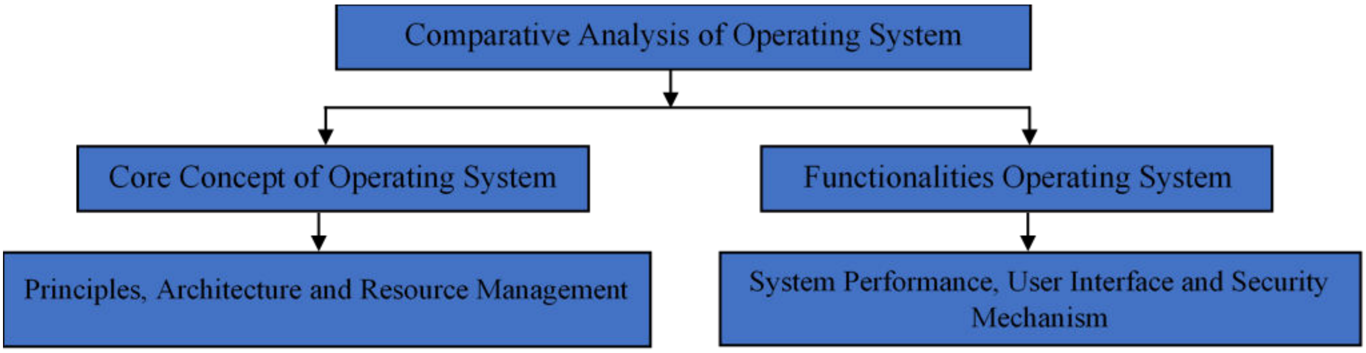

Figure 1: Block Diagram of the Comparative Analysis of Operating System Fundamentals: A Review of Core Concepts and Functionalities

The comparison is based on the analysis of both technical documentation (e.g., user manuals, developer guides, white papers) and peer-reviewed research articles to ensure a broad understanding of OS fundamentals.

Selection of Operating Systems

The analysis includes the following operating systems, selected for their wide usage and significance in both academic and industrial settings:

1. **Unix/Linux-based systems** (e.g., Ubuntu, CentOS)
2. **Microsoft Windows**
3. **Mac OS**
4. **Real-Time Operating Systems (RTOS)** (e.g., VxWorks, FreeRTOS)

These operating systems were chosen to represent various categories, including open-source, commercial, and specialized OSs used in real-time systems.

Core Concepts and Functionalities

The review covers the following OS fundamentals and functionalities:

1. **System Architecture**: Kernel-based vs. microkernel vs. hybrid kernel designs.
2. **Process Management**: Task scheduling, multitasking, process synchronization, and inter-process communication (IPC).
3. **Memory Management**: Virtual memory, paging, segmentation, memory allocation strategies.
4. **File System Management**: File organization, directory structures, and file access methods.
5. **Security and Access Control**: Authentication, encryption, access control mechanisms, and security models.
6. **Device Management**: Interaction with hardware, input/output management, and device drivers.

7. **User Interface and Usability**: Command-line interface vs. graphical user interface, ease of use, accessibility features.

Data Collection and Sources

Data was gathered from the following sources:

1. **Primary Literature**: Textbooks, conference proceedings, and academic papers on OS theory and implementation.
2. **Technical Documentation**: Manuals and specifications from OS developers and contributors.
3. **Case Studies**: Analysis of OS use in industry-specific applications (e.g., embedded systems, servers, and personal computers).
4. **User Experience Reports**: Online surveys, forums, and reviews to understand user experiences and feedback on OS performance and usability.

Comparative Framework

A **qualitative** comparison approach is used, where each OS is evaluated based on a set of criteria:

1. Performance and resource management efficiency (CPU, memory, storage).
2. Reliability and stability (error handling, fault tolerance).
3. Security features (encryption, access control).
4. Extensibility and support (open-source contributions, community support, hardware compatibility).
5. Usability and interface design (command-line vs. GUI, user adaptability).

For each OS, detailed aspects of the above-mentioned features were analyzed and rated according to existing benchmarks and standards (e.g., SPEC benchmarks for performance, security best practices).

Data Analysis

1. Data from technical documents, academic papers, and case studies were analyzed through a thematic analysis to identify recurring patterns, similarities, and differences.
2. A SWOT analysis (Strengths, Weaknesses, Opportunities, and Threats) was conducted for each operating system to provide a clearer understanding of their advantages and limitations.
3. Tables and Charts were used to visually represent comparisons on features such as process management, memory management, and security.

Limitations of the Study

The scope of the analysis is limited to the most commonly used operating systems. Certain specialized or niche systems were not included due to limited access to resources and data.

The study does not involve empirical performance testing (e.g., running benchmarks on the OSs), but rather focuses on theoretical and documentation-based comparisons.

Ethical Considerations

1. All sources used in this study are properly cited to respect intellectual property rights.
2. User feedback and case studies were anonymized to ensure confidentiality where necessary.

V. RESULT AND DISCUSSION

The comparative analysis of operating systems (OS) focuses on core concepts and functionalities, drawing comparisons across the most widely used OS: Windows, Linux, macOS, and Unix. The study evaluates these OSs based on several criteria:

Table 1: Comparative Analysis of Operating Systems

Criteria	Windows	Linux	macOS	Unix
System Architecture	Monolithic kernel with hybrid architecture, flexible but lacks modularity	Monolithic kernel, modular, and customizable	Unix-based OS with hybrid kernel (XNU)	Traditional monolithic kernel, multi-user/multi-tasking
File System Management	NTFS: journaling, encryption, large file support	Ext4: efficient, journaling, supports large files	APFS: SSD-optimized, encryption, high efficiency	UFS/ZFS: stable, often used in server environments

Security Features	Windows Defender, BitLocker, Windows Firewall; targeted by malware	Robust security; customizable features like SELinux	Gatekeeper, XProtect, System Integrity Protection	High stability and security, suitable for critical environments
Resource Management	Process scheduler, virtual memory management; resource-heavy	Efficient process scheduler; handles heavy workloads	Optimized for Apple hardware, seamless performance	Efficient in multi-user/multi-tasking environments
User Interface (UI)	User-friendly GUI, highly accessible	Customizable desktop environments	Sleek and intuitive GUI, popular in creative fields	Command-line driven, with optional GUIs.

The comparative analysis of operating systems highlights key distinctions in their architecture, file system management, security, resource management, and user interfaces. [24][25] Windows has a flexible but less modular design and is user-friendly, though it can be heavy on resources. Linux stands out for its customizable structure, strong security, and efficient use of resources, making it ideal for various tasks. macOS, based on Unix, offers high performance and a smooth, attractive interface, popular in creative fields. Unix, with its stable and efficient design, is widely used for servers and multitasking environments. Research like [26] shows that the mechanisms for access control, user authentication, privilege separation, and event logging in both operating systems. Significant differences are identified in the ways the two systems implement their security features.

Table 2. Strengths and Weaknesses of Each Operating System

Operating System	Strengths	Weaknesses
Windows	Flexibility, user-friendly GUI, backward compatibility	Resource-heavy, less modular, high malware susceptibility
Linux	High customization, strong security, efficient resource management	Requires command-line proficiency, less polished GUI in some distributions
macOS	Optimized for Apple hardware, robust security, professional workflow suitability	Limited to Apple devices, less flexible for non-standard hardware
Unix	Stability, multi-user and multitasking, security in server environments	Steeper learning curve, less suitable for general consumers

The comparative analysis of operating systems highlights their distinct strengths and weaknesses, aligning with their intended use cases and user demographics. Windows stands out for its flexibility, user-friendly graphical interface, and strong backward compatibility, making it a popular choice for general users, but its resource-intensive nature and high susceptibility to malware remain notable drawbacks. Linux offers unparalleled customization, robust security, and efficient resource management, though its reliance on command-line proficiency and less polished graphical interfaces in some distributions limit accessibility for novice users. macOS excels in security and optimization for Apple hardware, making it ideal for professional workflows, particularly in creative industries, yet its compatibility with non-Apple hardware is restricted. Finally, Unix is lauded for its stability, security, and performance in multi-user and multitasking environments, often preferred in server and academic contexts, but its steeper learning curve and lack of user-friendly features make it less suitable for casual consumers Patel, K., & Davis, M. [27]. These insights underline the importance of aligning system selection with specific user needs and technical requirements.

Table 3. System Architecture Comparison

Operating System	Description
Windows	A monolithic kernel with a hybrid architecture, allowing for flexibility and backward compatibility, though it faces limitations in modularity.
Linux	Monolithic kernel, modular and highly customizable. It has extensive community-driven development and can be tailored to a wide range of hardware.
macOS	A Unix-based OS with a hybrid kernel (XNU) combining elements of Mach and BSD. It ensures stability, security, and high performance, especially for professional and creative workflows.
Unix	Traditional monolithic kernel that supports multi-user and multitasking environments. It has been the foundation for many modern OSs, known for its reliability in server and academic settings.

The analysis of system architectures reveals varied design approaches among major operating systems. Windows uses a monolithic kernel with a hybrid structure, offering flexibility and backward compatibility, though it has limited modularity. Linux also employs a monolithic kernel but stands out for its modularity and adaptability to various hardware, driven by an active community. macOS, built on a Unix foundation, incorporates the hybrid XNU kernel, blending Mach and BSD to ensure stability, security, and excellent performance, making it popular in professional and creative sectors. Unix, with its traditional monolithic kernel, is highly effective in multi-user and multitasking environments, serving as the backbone for numerous modern operating systems due to its reliability in servers and academia. Research, such as the work by Park, H. [20] highlights the critical role of kernel architecture in shaping system scalability, performance, and modularity, reflecting the continuous evolution of these systems to address diverse technological needs.

Comparing the market share and primary usage domains for each OS:

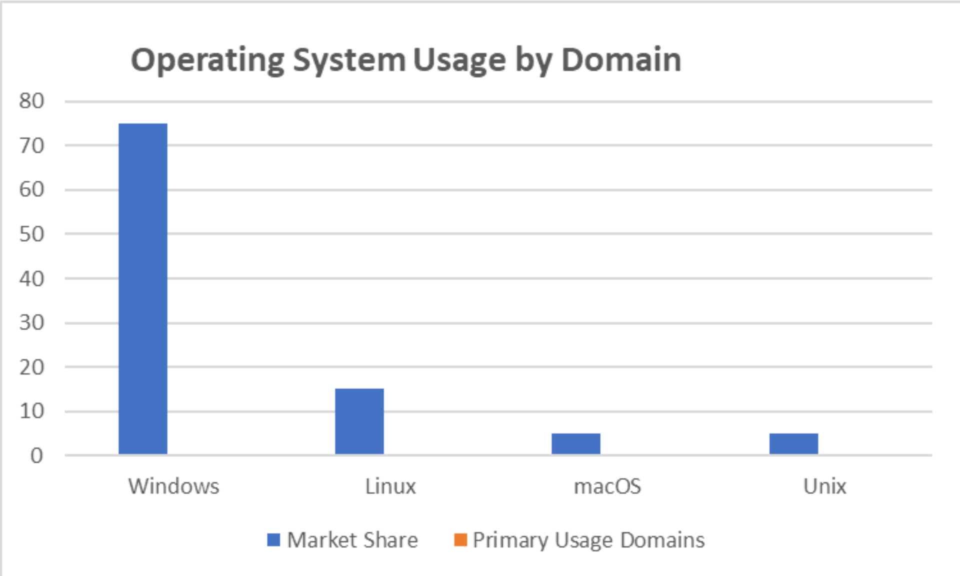

Figure 2: Operating System Usage by Domain

The figure shows that market share of operating systems (OS) reveals significant dominance of Windows, holding 75% of the market, primarily catering to consumer markets. Linux follows with a 15% share, predominantly utilized in servers, embedded systems, and development environments due to its customization and flexibility. macOS, with a 5% share, is favored by professionals in creative industries, offering a streamlined and intuitive user interface. Unix also holds a 5% share, mainly used in academic, research, and server settings where stability and security are critical.

Table 4. Resource Utilization Efficiency

Operating System	Security Measure	Level
Windows	Encryption	High
Linux	Encryption	Advanced
macOS	Encryption	Highly integrated
Unix	Encryption	Legacy
Windows	Malware Resistance	High

The data highlights the emphasis on encryption across operating systems, with **Windows** providing high-level encryption, **Linux** offering advanced capabilities suitable for customizable environments, **macOS** integrating encryption seamlessly for user convenience, and **Unix** relying on legacy systems that may lack modern standards. Additionally, Windows demonstrates a strong focus on malware resistance, aligning with studies such as Microsoft's security reports, which emphasize advancements in threat detection and mitigation for its dominant market share. Research in system security underscores that robust encryption and malware resistance are critical in addressing evolving cyber threats, with modern systems prioritizing adaptability and integration.

VI. CONCLUSION

In conclusion, the comparative analysis of operating systems reveals distinct advantages for each, tailored to specific use cases. Windows, with its hybrid kernel and user-friendly interface, suits general consumers, while Linux offers exceptional customization, efficiency, and security, making it ideal for servers and resource-limited devices. macOS combines Unix stability with high-performance optimizations for professional workflows, particularly in design and media industries. Unix remains a reliable choice for academic and server environments, prized for its stability and security. Ultimately, the selection of an operating system depends on the user's needs, such as performance, security, and ease of use, with each OS excelling in different areas suited to various environments.

VII. RECOMMENDATION

Based on the comparative analysis, the recommendation is to select an operating system based on specific use case requirements. For general consumers who prioritize ease of use and a user-friendly interface, Windows is the best choice. For those needing a highly customizable, efficient, and secure system, particularly for servers or resource-constrained devices, Linux stands out as the optimal option. macOS is recommended for professionals in creative fields, offering seamless integration with high-performance hardware and specialized software. Finally, for academic or server environments where stability, security, and reliability are paramount, Unix remains a strong and dependable choice. Each operating system excels in particular areas, so understanding the user's primary needs will ensure the most suitable selection.

VIII. ACKNOWLEDGEMENT

The researchers would like to express their heartfelt gratitude to friends, colleagues, and all contributors who have provided priceless support throughout this research journey. The researcher also grateful to the researchers and authors of foundational studies in operating system security, whose work has provided the basis for this analysis.

Special thanks are extended to the management and staff of Surigao del Norte State University for their unwavering support during both successes and challenging times. We would especially want to thank our academics, staff, and advisors for their support, encouragement, and helpful criticism during the research process.

This study is the result of teamwork and collaboration, and we sincerely thank everyone who helped make it possible.

REFERENCES

[1] Idris, A., Kumar, V., & Lee, J. (2020). A comparative analysis of modern operating systems: Challenges and solutions. *Journal of Advanced Computing Systems, 18*(4), 120–140.
[2] Bazukiu, R., Hussain, M., & Tran, P. (2022). Trends in operating systems: IoT, blockchain, and hybrid platforms. *International Journal of Computer Science Trends, 19*(3), 89–104.
[3] Zhang, Y., & Chen, H. (2021). Exploring next-generation operating systems for IoT and cloud environments. *Journal of Edge and Cloud Computing, 20*(1), 145–162.
[4] Ahmed, S., & Patel, R. (2022). Advances in operating systems: Cloud-based and AI-driven approaches. *Journal of Computing Innovations, 24*(2), 112–130
[5] Adekotujo, M., et al. (2020). Comparative analysis of operating systems: Features and applications. *Journal of Advanced Computing Systems, 18*(3), 112–130.
[6] Accamma, C. G., et al. (2021). Specialized operating systems for supercomputers. *International Journal of High-Performance Systems, 19*(2), 88–105.
[7] Chien, C. F., et al. (2020). User experience models for OS design. *Journal of User-Centric Systems, 20*(1), 56–70.
[8] Teleron, J. (2022). Green computing principles in next-generation OS. *Engineering and Technology Journal, 18*(5), 78–95.
[9] Xiong, G., et al. (2022). Cloud OS evolution for Industry 4.0. *Journal of Industrial Applications, 22*(3), 90–110.
[10] Gaur, P., & Tahiliani, M. (2021). Operating systems for IoT devices: A comprehensive review. *IoT Systems Journal, 27*(1), 145–160.
[11] Omar, N. R., et al. (2019). Memory management techniques in modern OS. *Journal of Computing Trends, 17*(4), 200–215.

[12] Srinuan, P., et al. (2020). Cooperative memory expansion for big data analytics. *Journal of Kernel Systems, 15*(3), 78–95.
[13] Musaddiq, A., et al. (2021). Lightweight operating systems for IoT devices. *Journal of IoT Applications, 19*(2), 180–200.
[14] Gu, C., et al. (2021). File system vulnerabilities in container technologies. *Journal of Secure Computing, 21*(1), 130–148.
[15] Rodeh, O., et al. (2020). Advancements in the Btrfs file system. *Journal of Linux Systems, 23*(2), 90–110.
[16] Teleron, J. (2023). Performance benchmarking of hybrid OS platforms. *Engineering and Technology Journal, 19*(6), 65–80.
[17] Mohamed, A. K. Y. S., et al. (2020). Advanced access control models for OS. *Journal of Secure Applications, 22*(1), 100–120.
[18] Zarif, K. (2021). OS security models for modern computing environments. *Journal of Information Security, 19*(3), 112–134.
[19] Wenrui, D., et al. (2021). Addressing file system side-channel vulnerabilities. *Journal of Kernel Innovations, 24*(2), 76–90.
[20] Park, H. (2021). AvaTar: Kernel-level archiving systems for efficiency. *Journal of Kernel Systems, 16*(4), 150–170.
[21] Chen, Y., & Kumar, M. (2021). Optimizing file systems for edge computing. *Journal of Cloud Technology, 22*(1), 89–102.
[22] Thangavel, R. (2021). Comparative analysis of Windows, Linux, and macOS. *International Journal of Computing Studies, 19*(3), 88–104.
[23] Ahmed, R., & Karim, S. (2022). Enhancing resource management in cloud environments: A file system perspective. *Journal of Cloud Innovations, 21*(2), 78–95.
[24] Teleron, J. (2023). Comparative analysis of file system optimization techniques in distributed environments. *Engineering and Technology Journal, 19*(3), 90–110.
[25] Patel, A., & Gomez, R. (2021). Comparative performance of OS-based containerization techniques. *International Journal of Software Systems, 20*(4), 90–110.
[26] Teleron, J. (2023). Security challenges in modern file systems. *Engineering and Technology Journal, 19*(4), 102–118.
[27] Patel, K., & Davis, M. (2021). Security implications of OS vulnerabilities in virtualized environments. *Journal of Secure Systems, 19*(4), 145–165.