M.ALI HURR

221325

BSCS 5 C

OPERATING SYSTEM

ASSIGNMENT 4

AIR UNIVERSITY

WHAT IS OS:

Introduction Operating systems (OS) are foundational software enabling the interface between hardware and users, offering a platform for application execution. This analysis compares Android (mobile) and macOS (desktop/laptop), focusing on their architecture, key functionalities, and differences in process management, memory management, file systems, security, and scheduling.

1)Process Management Android uses the Linux kernel for process management, relying on features like the Low Memory Killer (LMK) for resource-constrained environments. Processes are managed with specific priorities, leveraging mechanisms like JobScheduler for deferred execution and Binder for inter-process communication (IPC). Multitasking is effective, albeit optimized for mobile resources.

macOS uses the XNU kernel, a hybrid design combining microkernel and monolithic elements. It employs the Mach scheduler for dynamic task prioritization and robust IPC mechanisms, such as Mach messages, ensuring smooth multitasking. macOS offers a more complex scheduling system due to its broader application spectrum.

DIFFERENCEAndroid emphasizes power efficiency and limited resources, while macOS focuses on performance and scalability.

2) **Memory Management** Android employs Dalvik/ART runtime with garbage collection to manage memory. Virtual memory uses paging and swapping but optimizes for minimal overhead. It also has efficient caching and memory protection for application sandboxes.

MacOS provides advanced virtual memory through swapping and demand paging, backed by its tight hardware integration. APFS ensures robust file mapping, and caching systems enhance memory efficiency. Memory protection is rigorous, ensuring user-level isolation.

Difference: macOS offers a more expansive memory management model, while Android is tailored for mobile constraints.

3) <u>File System</u> Android predominantly uses the ext4 file system, supporting journaling and efficient space allocation. The system organizes data into user and system directories with strict access controls.

MacOS uses APFS (Apple File System), optimized for flash storage. APFS supports snapshots, encryption, and space sharing for modern storage devices.

<u>Difference :</u>APFS is advanced in security and efficiency, while ext4 is lightweight, suiting Android's needs.

4) **<u>Security</u>** Android incorporates sandboxing, SELinux policies, and regular security patches. Permissions are app-specific, and encryption standards like AES are used for user data.

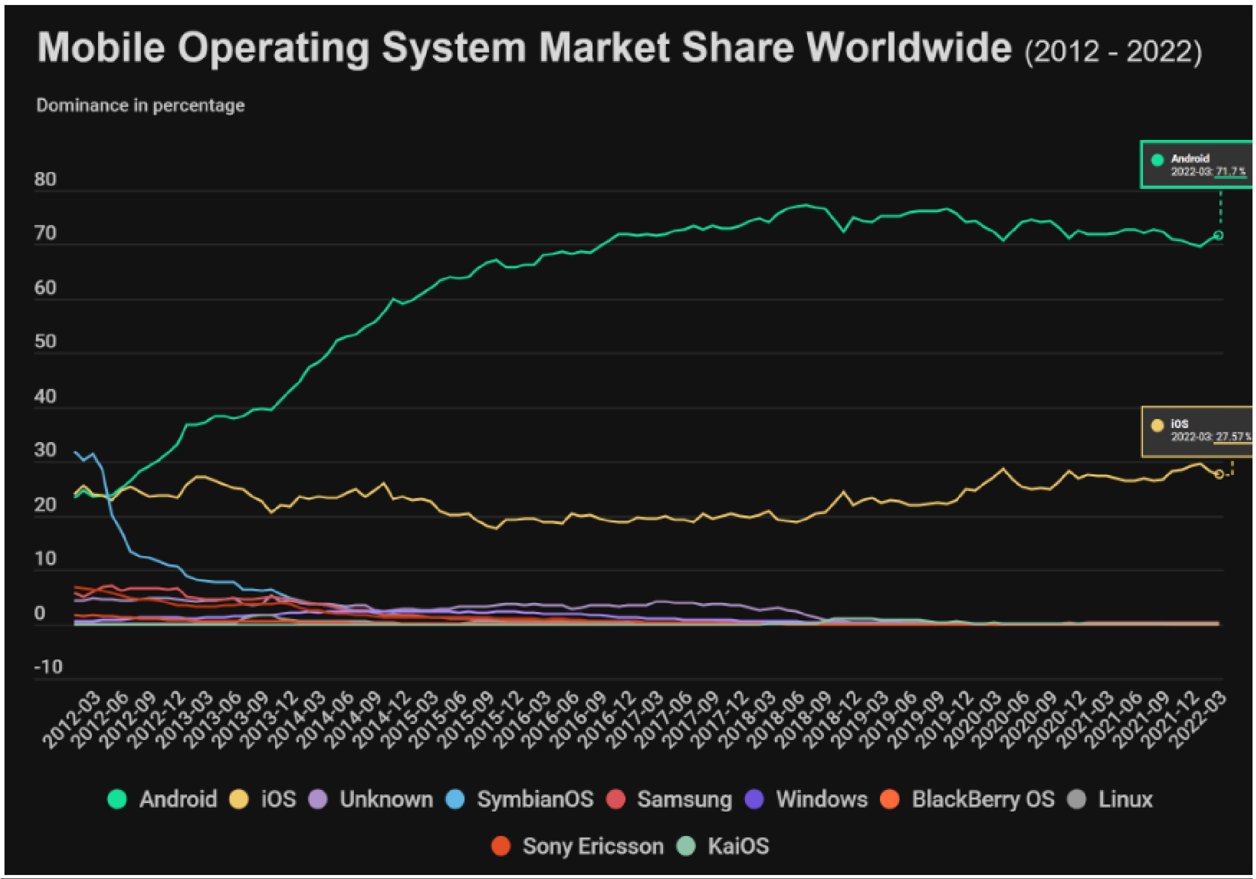
MacOS integrates encryption (FileVault), secure boot, and application notarization to prevent unauthorized software. It also uses kernel-level protections and advanced permissions systems.

<u>Difference:</u> macOS is stronger in comprehensive security, whereas Android focuses on securing a diverse ecosystem.

5) <u>Scheduling</u> Android's scheduling, based on Completely Fair Scheduler (CFS) from Linux, ensures fairness while managing mobile-specific constraints. Real-time scheduling is limited to media processes.

MacOS uses Mach scheduling, offering real-time process handling and dynamic thread priorities for multitasking and user responsiveness.

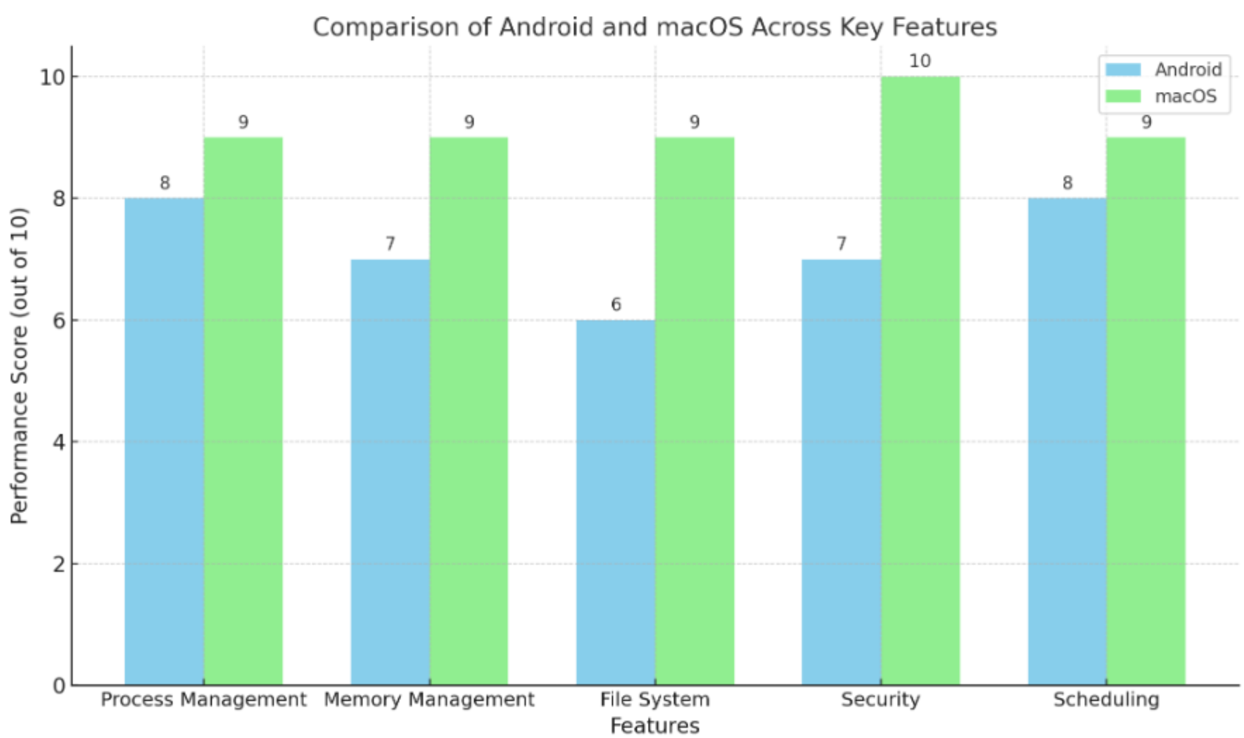
<u>Difference :</u>macOS provides superior real-time capabilities, suited for professional workloads.

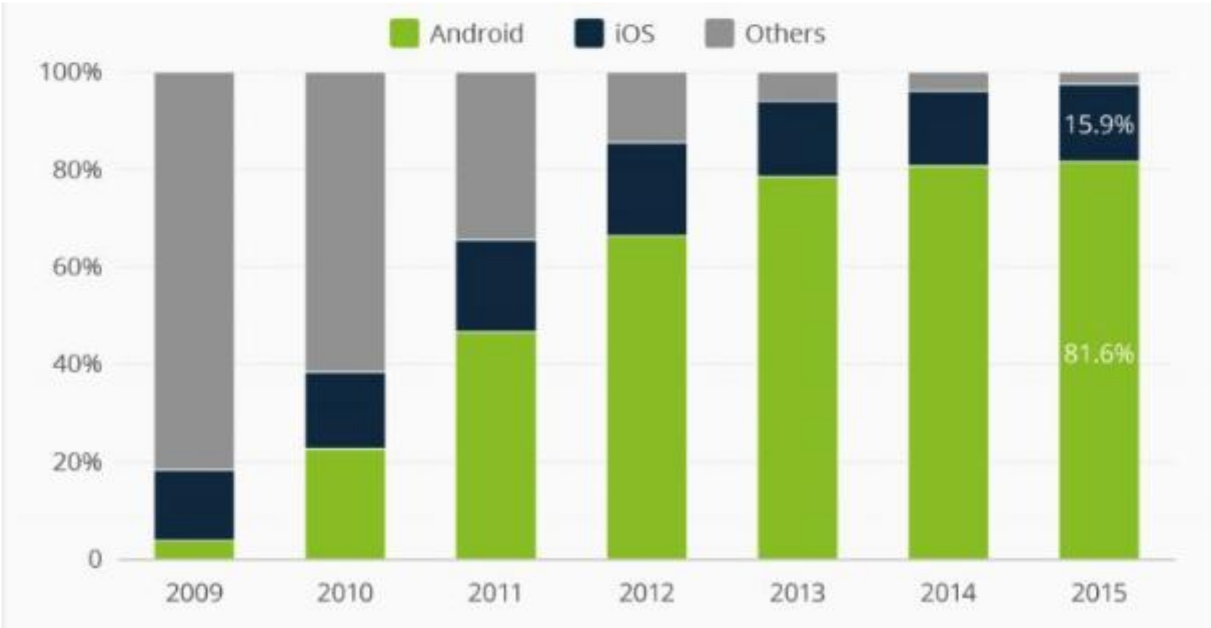


General comparion of osAndroid and macos: Android and macOS have distinct approaches to process management, reflecting their target environments. Android, optimized for mobile devices, uses the Linux kernel's capabilities to prioritize power efficiency and resource conservation. It employs the Low Memory Killer (LMK) to terminate non-critical processes when memory is scarce, focusing on seamless performance for active apps. The use of JobScheduler and Binder IPC ensures efficient multitasking and secure inter-process communication within the constraints of mobile hardware. In contrast, macOS leverages the XNU kernel, integrating elements of microkernel and monolithic designs to handle resource-heavy desktop applications. The Mach scheduler in macOS dynamically allocates CPU time based on priorities, providing superior multitasking for professional and creative workloads. Memory management further highlights their differences: Android uses Dalvik/ART runtimes with garbage collection for automatic memory handling, minimizing swapping to preserve battery life, whereas macOS employs advanced virtual memory mechanisms, including demand paging and robust caching, to ensure responsiveness under heavy loads. The file system architectures of Android and macOS demonstrate their alignment with device-specific needs. Android uses the ext4 file system, offering journaling and encryption at the block level to secure user data while maintaining lightweight performance suitable for mobile storage. On the other hand, macOS employs the Apple File System (APFS), which is optimized for SSDs and provides advanced features like snapshots, dynamic space allocation, and full-disk encryption, making it ideal for modern high-performance computing. Security is another key differentiator. Android emphasizes openness with mechanisms like sandboxing, SELinux policies, and app-specific permissions to mitigate risks in its diverse app ecosystem. macOS, however, benefits from tighter hardware-software integration, incorporating features like FileVault for disk encryption, secure boot, and app notarization to maintain a controlled and secure environment. While Android excels in flexibility and adaptability for a wide range of hardware, macOS prioritizes robust security and seamless performance, reflecting its closed ecosystem and professional user base.

Creative Analogies: Android and macOS, both robust operating systems, cater to different device types and user needs. Android's process management prioritizes efficiency and power conservation, using mechanisms like JobScheduler and Binder for multitasking and inter-process communication, while macOS's XNU kernel supports advanced multitasking through the Mach scheduler and robust IPC. In memory management, Android employs runtime garbage collection and minimal swapping, optimized for mobile resource constraints, whereas macOS uses demand paging and tight hardware integration for seamless performance.

Android’s ext4 file system ensures efficient storage with block-level encryption, while macOS’s APFS offers advanced features like snapshots and dynamic space sharing, enhancing security and usability. Security-wise, Android relies on sandboxing and SELinux policies, whereas macOS uses FileVault encryption, secure boot, and stringent application notarization. Finally, Android’s Linux-based Completely Fair Scheduler balances fairness and energy efficiency, while macOS’s dynamic scheduling ensures smooth performance even under heavy workloads, reflecting their tailored approaches to mobile and desktop environments.





RESEARCH PAPER LINK

https://www.researchgate.net/profile/Adedoyin-Odumabo/publication/372400705_A_Comparative_Study_of_Operating_Systems_Case_of_Windows_UNIX_Linux_Mac_Android_and_iOS/links/64b41d62c41fb852dd7b65e1/A-Comparative-Study-of-Operating-Systems-Case-of-Windows-UNIX-Linux-Mac-Android-and-iOS.pdf

https://www.researchgate.net/profile/Nimesh-Ekanayake/publication/325257105_Android_Operating_System/links/5d39e59ba6fdcc370a603bc2/Android-Operating-System.pdf?origin=publication_detail&_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI

6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uRG93bmxvYWQiLCJwcmV2aW91c1BhZ2UiOiJwdWJsaWNhdGlvbiJ9fQ

https://www.researchgate.net/publication/342625092_Development_of_iOS_A_Revolutionary_Transformation_and_the_Future/link/5f019851299bf18816037b59/download?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19

git repo link

https://github.com/malihurr/os/upload/main