# **Assignment 3**



Name: Muhammad Zubair Khan

Class: BSCS-V-C

ID: 221415

Subject: Operating System

Submitted to: Mam Warda Aslam

# **MAC OS vs Android OS Comparison**

# **Process Management**

## **Android OS:**

* **Foundation:** A Linux-based kernel the powers Android's process management, which uses the Completely Fair Scheduler (CFS) to distribute resources.
* **Task Handling:** Uses the Zygote process to reduce overhead by forking existing processes instead than starting from scratch uses the Zygote process to reduce overhead by forking existing processes instead than starting from scratch.
* **Background Task:** Implements a Background Task Manager to prioritize foreground processes and manage resources efficiently.
* **Openness:** Gives developers flexibility, but also has drawbacks like fragmentation and heightened susceptibility.

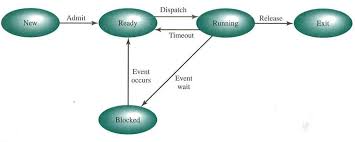


Figure 1: The Process Management in Android OS

## **Mac OS:**

* **Foundation:** With a focus on stability and security, macOS employs a hybrid kernel that combines aspects of BSD and Mach.
* **Task Handling:** Maximizes system responsiveness by using Grand Central Dispatch (GCD) for parallel job execution.
* **Background Execution:** Properly balances power consumption and performance when handling resource-intensive tasks.
* **Ecosystem:** Although macOS is a restricted to closed-source system, it guarantees consistent user experience and efficient speed, yet also restricts developer flexibility.

**Comparison:** Android, built on the Linux kernel, offers flexibility but faces security and fragmentation issues, using the Zygote process to reduce overhead. In contrast, macOS, with its hybrid BSD-Mach kernel, prioritizes security and stability within a closed ecosystem, using Grand Central Dispatch (GCD) for efficient multitasking. Android relies on a Background Task Manager to optimize resources.

# **Memory Management**

## **Android OS:**

* **Approach:** Optimizes memory allocation and shortens application launch time by integrating the Zygote process and Android Runtime (ART).
* **Efficiency:** Memory performance was enhanced by switching from Dalvik Virtual Machine (DVM) to ART via Ahead-of-Time (AOT) compilation.
* **Challenges:** The different hardware configurations cause fragmentation problems, which result in inefficient memory use. Performance optimization across systems with different architectures may be challenging as a result of these variations.

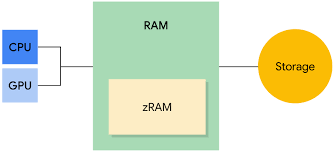


Figure 2: Memory Management in Android OS

## **Mac OS:**

* **Approach:** By allocating resources as needed while a program is running, dynamic memory allocation aids with memory management. By using virtual memory approaches, the system can optimize resource management by using disk space as an extension of RAM.
* **Isolation:** Sandboxing keeps apps from accessing vital system resources by isolating them in a regulated setting. This keeps possible threats inside the isolated manipulate, ensuring increased security and stability.
* **Performance:** By lowering the amount of memory needed for high-performance operations, optimized memory compression increases productivity. Better memory management is made possible by paging algorithms, which switch data between RAM and the disk as needed for quick execution.

**Comparison:** The Zygote process and ART in Android OS increase memory efficiency, yet fragmentation brought on by different hardware can make performance tuning more difficult. For increased security and stability, macOS makes use of virtual memory, dynamic memory allocation, and sandboxing. It also optimizes memory compression and paging methods for improved efficiency. All things considered, macOS provides more reliable memory management, whereas Android struggles with hardware fragmentation.

# **File System**

## **Android OS:**

* **Structure:** The ext4 file system is commonly used for mobile storage devices due to its performance and reliability. For external storage, FAT32 and exFAT are favored for their wide compatibility and support for large file sizes.
* **Access:** The open file system makes data management and transfer simple by enabling access via USB or system-level tools. Because it is tailored for app-specific storage, app-related data may be efficiently organized and retrieved.

## **Mac OS:**

* **Structure:** Advanced capabilities like cloning for effective file copying and snapshots for data backups are provided by APFS (Apple File System). Additionally, it offers robust encryption, which improves data security on Apple devices.
* **Access:** On Apple devices, file handling and navigation are made easier by the user-friendly graphical user interface. It offers a dependable and effective user experience by blending in well with Apple's environment.

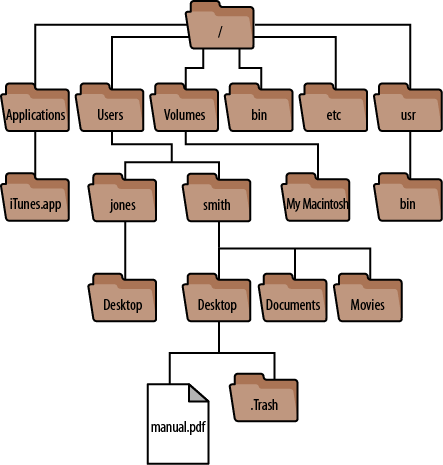


Figure 3: File System in Mac OS Terminal

**Comparison:** With an open file system that makes data management easier, particularly for app-specific storage, Android OS prioritizes performance, dependability, and compatibility. It uses FAT32/exFAT for external storage and ext4 for mobile storage. On the other hand, macOS makes use of the sophisticated APFS, which provides capabilities like snapshots, cloning, and strong encryption for improved file management and data protection. Additionally, macOS offers an intuitive graphical user interface for easy navigation, guaranteeing a seamless integration into the Apple ecosystem. Android places more emphasis on compatibility and flexibility, but macOS is superior at advanced file handling and security.

# **Security**

## **Android OS:**

* **Permissions:** Users can regulate the data and resources that each app can access using granular app-level permissions. By limiting needless app rights, this improves security and privacy.
* **Encryption:** By using strong encryption methods to safeguard files, AES-based file-level encryption guarantees safe data storage. This method stops unwanted access and improves data privacy.
* **Authentication:** Face unlock, and fingerprint scanning are two examples of biometric authentication that provide a safe and practical way to log in. For more flexibility, it complements more conventional choices like passwords and PINs.

## **Mac OS:**

* **Permissions:** In order to regulate access to files and directories according to user roles, Unix-style file permissions are enforced. These contain permissions for owners, organizations, and others to read, write, and execute. Strong security and effective resource management are guaranteed by this framework.
* **Encryption:** Complete full-disk encryption is provided by FileVault integration, protecting all stored data. To stop unwanted access to private data, it encrypts the entire disk. Users of the system will benefit from increased security and privacy as a result.
* **Authentication:** Multi-factor authentication is supported by the system to provide an additional degree of protection over standard passwords. It uses encryption keys at the system level to protect data at the fundamental level. When taken as a whole, these safeguards provide strong defense against unwanted access.

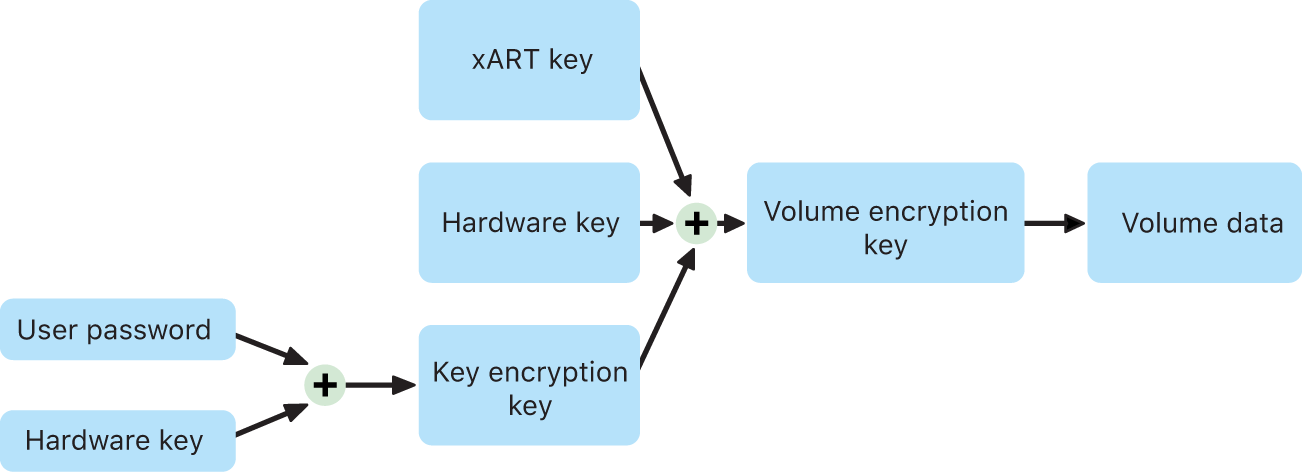


Figure 4: File Vault in Mac OS

**Comparison:** In addition to AES-based file encryption to protect data, Android OS provides granular app-level permissions for enhanced security and privacy. It enhances conventional login techniques by supporting biometric authentication such as fingerprint scanning and face unlock. In contrast, macOS employs FileVault for full-disk encryption and Unix-style file permissions for strong access control, guaranteeing complete data protection. In comparison to Android, it offers a higher degree of security and privacy by supporting multi-factor authentication with encryption keys.

# **Scheduling**

## **Android OS:**

* **Algorithm:** To distribute CPU time among processes equitably, the Completely Fair Scheduler (CFS) is used. By ranking jobs according to their virtual runtime, it guarantees fair resource distribution. This method keeps the system responsive while increasing efficiency.
* **Prioritization:** To reduce their influence on system performance, background tasks are given a lower priority. This guarantees that important tasks are given enough resources to be completed effectively. The method improves the overall stability and responsiveness of the system.
* **Challenges:** Addressing the changing needs of complex applications requires constant improvement. In dynamic contexts, compatibility, performance, and dependability are guaranteed by frequent upgrades. Both user satisfaction and system efficiency are preserved by this iterative process.

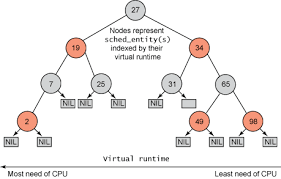


Figure 5: Completely Fair Scheduler

## **Mac OS:**

* **Algorithm:** Effective task execution management and CPU resource allocation are achieved with the Mach Scheduler. Grand Central Dispatch (GCD) optimizes load balancing and work allocation across numerous cores to further increase efficiency. In multitasking contexts, this combination guarantees responsiveness and seamless performance.
* **Real-time Tasks:** In order to prioritize tasks that require urgent attention and ensure timely execution, the system incorporates real-time scheduling. For apps that interact with users, interactive scheduling further maximizes performance while preserving responsiveness. For important and participatory procedures, this method reduces delays.
* **Efficiency:** Because of its multitasking optimization, the system can manage several processes well without sacrificing performance. In order to guarantee seamless user interactions and task transitions, responsiveness is given top priority. The system's overall efficiency is improved by this emphasis on user pleasure.

**Comparison:** To guarantee fair CPU time distribution, Android OS prioritizes foreground processes for system performance using the Completely Fair Scheduler (CFS). In order to meet the changing requirements of complex applications, it emphasizes incremental improvements. On the other hand, macOS makes use of the Mach Scheduler with Grand Central Dispatch (GCD) to prioritize urgent jobs in real-time scheduling and optimize load balancing across several cores. macOS places a strong emphasis on speed and multitasking efficiency to guarantee seamless user engagement and performance.

# **Analogy: Two Different Neighbourhoods**

**AndroidOS (The Creative Neighbourhood):** Like a lively neighbourhood, Android is teeming with inventors and artists. Street concerts, pop-up stores, and vibrant murals abound, and anybody may put up a booth (app) to display their artwork. Although this encourages diversity and innovation, it can also feel disorganized. Finding a dubious vendor selling fake items (virus) is a possibility, and the arrangement might be confusing because every booth has a unique design.

**MacOS (The Organized Community):** In contrast, macOS resembles a well-kept community with lovely gardens and well-maintained residences. Clear regulations are in place to preserve security and order, and everything is made to be comfortable and effective. People in this community like a calm, reliable environment where everything works together effortlessly, even though it may feel a little restrictive at times. Residents feel protected and secure because of the robust security that keeps viruses and other unwelcome guests out.

# **Insights and Observations**

**Flexibility vs Security:** You may modify and explore a lot with Android, but there may be security risks. On the other hand, macOS provides a secure and reliable environment, but with limited customization possibilities.

**User Experience:** Due to the variety of devices and manufacturers, Android may offer a wide range of experiences. While some might be excellent, others might not be. Users find it easier because macOS offers a consistent experience across devices.

**Innovation vs Stability:** Android promotes creativity, enabling programmers to produce original and captivating applications. But occasionally, this can result in errors or poor performance. Stability is a top priority for macOS, so even if it's not always the newest technology, everything works properly.

# **Personal Reflections**

I appreciate Android's innovation and flexibility as a user, but I also appreciate macOS's seamless and safe interface. I find myself drawn to either one depending on what I need, be it dependability or customisation. In the end, each operating system has advantages and disadvantages that represent various perspectives on technology and user requirements.

# References

*ijert*. (2023, November 11). From <https://www.ijert.org/research/process-management-in-android-and-ios-IJERTV12IS110230.pdf?>

*research gate*. (2020, July). From <https://www.researchgate.net/publication/372400705_A_Comparative_Study_of_Operating_Systems_Case_of_Windows_UNIX_Linux_Mac_Android_and_iOS>