**Course Name: - Operating System**

**A blue logo with a bird

Description automatically generated Assignment # 03**

**Submitted By: -**

**Alishba Fatima (221352)**

**Submitted To: -**

**Mam Warda Aslam**

**Department of Computer Science**

**Air University, Islamabad**

***Analysis and Comparison of Mobile OS (Android/iOS) and macOS:***

***1. Introduction:***

I will be analyzing and comparing the operating systems Android, a popular mobile OS, and Apple's macOS, used for desktops and laptops. The main focus is to explore architectural and functional details of each system based on key OS concepts, such as process management, memory management, file system, security, and scheduling.

***2. Research Papers:***

For me to understand the technicalities of both operating systems, I referred to some academic research papers and industry white papers found on Google Scholar, IEEE Xplore, and the ACM Digital Library.

***3. Summary of Research Papers:***

**Summary of Android Research Paper:**

Research in the Android operating system delves deep into architecture that considers a discussion in design and implementation of crucial components, including process management, memory management, and security. It describes how Android, built on top of the Linux kernel, provides flexibility and customizability of mobile applications.

***Process Management:*** The paper discusses how Android handles process creation through the Zygote process, which preloads shared libraries for faster app startup. It also highlights Android's use of Binder for inter-process communication (IPC), ensuring that processes can securely share data and interact efficiently.

***Memory Management:*** The paper stresses Android's garbage collection mechanism and virtual memory management, highlighting how the OS runs DALVIK/ART-based execution, which allocates, deallocates memory during process, and thus assures one gets the best with restricted mobile resources.

***Security:*** On permission-based model about applications, the paper discussed which application needs to attain one's permission before receiving it for access to sensitive data. About using encryption and authentication method regarding Android device security.

***Scheduling:*** Android CPU scheduling mechanisms are also covered, especially how the OS allocates the CPU resources depending on whether the tasks are in foreground or background.

The **conclusion** about Android is that it indeed achieves a balance between its performance and flexibility but presents a security risk as a free, open-source product; its developers must address it.

**Summary of Mac OS Research Paper:**

The research on macOS gives an in-depth analysis of the operating system's Unix-based architecture and its application in desktop and laptop environments. It further explains how macOS is optimized for high performance and security, especially for professional and creative work.

***Process Management:*** The paper explains how macOS utilizes the Darwin kernel, a hybrid kernel that combines elements of Mach and BSD Unix, to manage processes and threads. It elaborates on how the Kernel schedules processes and provides multitasking support through preemptive scheduling.

***Memory Management:*** The paper points out advanced memory management techniques used in macOS, such as paging, virtual memory, and memory protection. It discusses how it can separate applications in virtual memory spaces, thus eliminating unauthorized access to memory.

***Security:*** macOS is known for its robust security mechanisms that include sandboxing, full disk encryption (FileVault), and two-factor authentication. The paper analyzes how macOS enforces app security through strict permissions and system integrity checks.

***Scheduling:*** The research paper delves into how macOS's Grand Central Dispatch (GCD) system dynamically schedules tasks across multiple cores, optimizing performance for multi-threaded applications. It also goes deep into real-time scheduling for media-related tasks.

The paper **concludes** that macOS is meant to be used professionally and thus has been designed for high performance, security, and stability, making it the best-suited operating system for environments like content creation, programming, and system administration.

***4. OS Concepts Comparison:***

**Process Management:**

***Android:***

***Process Creation:*** In Android, processes are created through the Android application framework, which interacts with the Linux kernel. When an app is launched, it is placed into a process that is managed by the Zygote process (which creates child processes for each application).

***Scheduling:*** Android uses the Linux kernel scheduler for multitasking. It uses preemptive scheduling, which means that tasks can be interrupted to switch to another task. Android has a system called Activity Manager that manages the lifecycle of applications and multitasking.

***Inter-Process Communication (IPC):*** Android uses Binder, a high-performance IPC mechanism for communication between different processes. It allows data sharing and method calls between applications.

***MacOS:***

***Process Creation:*** macOS, being based on Darwin, which is a Unix-based OS, creates processes using the standard Unix system calls. Whenever an application is launched, a new process is created under the control of the kernel.

***Scheduling:*** macOS uses a preemptive multitasking model, just like Unix. The kernel scheduler allocates CPU time for various processes based on their priority levels. It also supports thread-level multitasking.

***IPC:*** macOS uses several IPC mechanisms, including Mach ports, which enable processes to communicate. It also supports Apple's XPC (Cross-Process Communication) for more modern application interaction.

**Memory Management:**

***Android:***

***Memory Allocation/Deallocation:*** Android manages memory through its Virtual Machine (Dalvik/ART), where memory is allocated for apps and deallocated when no longer needed. The garbage collector frees up memory by removing unused objects.

***Virtual Memory & Protection:*** Android uses a Linux kernel to manage memory. The memory used in this device is virtual. Every application runs in its isolated process with its own address space.

***Caching:*** Android keeps application data in RAM and storage for improved app startup times.

***MacOS:***

***Memory Allocation/Deallocation:*** macOS takes the more traditional approach by the Unix kernel. It allocates memory through a combination of paging and virtual memory and, upon process exit, deallocates memory automatically.

***Virtual Memory & Protection:*** The macOS provides robust virtual memory management, ensuring that every application has its own address space. It uses sophisticated mechanisms of memory protection, thereby preventing applications from accessing each other's memory.

***Caching:*** macOS uses multiple types of caching, including file system and application-level caches, to improve performance.

**File System:**

***Android:***

***File Storage:*** Android primarily uses the ext4 file system for internal storage. For external storage, it relies on FAT32 or exFAT file systems, especially on SD cards.

***Organization:*** The file system on Android is organized in a hierarchical manner, with directories like /system, /data, and /cache. Each application has its sandboxed storage.

***MacOS:***

***File Storage:*** macOS uses APFS (Apple File System) for both internal and external storage. APFS provides more data integrity, encryption, and performance than its predecessor HFS+.

***Organization:*** The macOS file system is hierarchical, with standard directories like /Applications, /Users, /Library. Files are organized into bundles, where the application data and resources are grouped.

**Security:**

***Android:***

***Permissions:*** Android needs apps to request permissions before accessing any sensitive data such as contacts, location, and camera. All permissions are managed by the system and can be taken back by the user at any time.

***Encryption:*** Android supports full-disk encryption, which protects data in the device. The user is not aware of how the encryption process is occurring.

***Authentication:*** Android uses biometric authentication (fingerprint, face recognition) and PIN/password-based authentication for unlocking the device.

***MacOS:***

***Permissions:*** macOS employs the sandboxing technique, which prevents applications from accessing unauthorized data or system resources. Applications also need permission from the user to access sensitive data.

***Encryption:*** macOS uses FileVault for full disk encryption, so all files on the device are encrypted and protected.

***Authentication:*** macOS employs various authentication methods, including Apple's Touch ID, password-based authentication, and more robust methods like 2FA (Two-Factor Authentication) for iCloud and system settings.

**Scheduling:**

***Android:***

***CPU Scheduling:*** Android employs the Linux scheduler, which carries out time-sharing and priority-based scheduling. Applications that are currently active or in the foreground have higher priority.

***Real-time Processing:*** Android enables real-time processing for special applications, such as video streaming and gaming, by virtue of native libraries available through Android, such as OpenGL for rendering.

***MacOS:***

***CPU Scheduling:*** macOS leverages the GCD and Operation Queues for the smooth scheduling of tasks on multiple cores. It dynamically changes process priority and thread distribution

***Real-time Processing:*** Support for real-time processing with audio and video synchronization is supported by macOS, which offers a professional-level service.

***5.Creative Analogy:***

Consider Android as a city with many roads and vehicles. Each app is like a vehicle, and the Zygote process is like the traffic control system that ensures that all the vehicles are ready to go. The Android kernel is the city's police force, which enforces rules, schedules traffic, and allows smooth transitions. Meanwhile, macOS is like a structured university campus, where departments work independently but share resources - like libraries, computing facilities, and communication networks - underneath a strict administrative body or governing body (the Unix kernel). The university sets up everything in order and optimize things for the students and keeps each department's lines (virtual memory and file permissions) well defined.

***6. Insights and Personal Observations:***

**Android** is more flexible and customizable, allowing users to tailor the OS to their needs. It is optimized for mobile devices with considerations like battery life, power consumption, and inter-app communication.

**macOS**, on the other hand, offers a more polished and unified user experience, focused on high performance, data integrity, and security, making it suitable for professional workstations and creative applications.

While both operating systems are based on core Unix principles, Android is more open due to its reliance on Linux, whereas macOS's closed ecosystem provides tighter integration and security.

***7. Conclusion:***

Both Android and macOS have evolved along different lines of design philosophy, serving different user needs and environments. Android is more suitable for mobile devices in terms of flexibility and scalability, while macOS is designed to be used for performance, security, and a seamless user experience on desktop/laptop environments. This is an important consideration when choosing the right OS for specific applications or user requirements.