

# Process Management In Android And IOS

Sankalp Patil  
of E&TC,  
Pune Affiliated by SPPU,  
Pune, Maharashtra, India, 411048

Yash Patil  
dept. of E&TC,  
VIIT Pune Affiliated by SPPU,  
Pune, Maharashtra, India, 411048

Prof. Minal Deshmukh dept.  
dept. of E&TC,  
VIIT Pune Affiliated by SPPU  
Pune, Maharashtra, India, 411048

**Abstract:** Process management is essential to guarantee optimal user experience on the millions of mobile devices powered by the likes of Android and iOS. These two operating systems have become such fundamental components of our daily routines that it is crucial to understand their intricate mechanisms. By exploring the unique intricacies of process management in Android and iOS, this paper aims to shed light on how they impact device performance. Process management in the Android world is a task involving resource management, memory allocation, and task scheduling. Our investigation of these vital features offers insight into the latest Android updates and improvements. iOS, on the other hand, is a closed-source OS that utilizes a distinct approach to process management with a focus on optimizing resources, ensuring proper stability, and prioritizing security. Our detailed overview of iOS's process management policies allows readers to easily identify how it differs from Android in significant ways. User experience and system efficiency are both heavily impacted by the similarities and differences in process management between Android and iOS. A thorough comparison of these aspects sheds light on their implications, as well as on the necessity of understanding them. Managing increasing system complexity and optimizing power are just a couple of challenges facing both iOS and Android. Through research and proposed solutions, the future of mobile operating system process management will undoubtedly be shaped. This review paper takes a closer look at identifying these challenges and the potential for further investigation. Mobile technology researchers, developers, and policymakers can benefit from a thorough understanding of the intricacies involved in process management on Android and iOS devices. This review paper serves as a valuable resource by providing a comprehensive overview on how this role plays a pivotal role in determining device performance and user satisfaction.

## INTRODUCTION

With the help of mobile operating systems, digital technology has transformed how we interact with it. Smartphones and tablets seem like they've always been a part of our lives, but it wasn't that long ago when they weren't. Google's Android and Apple's iOS are two pivotal heads of this wave, orchestrating the experiences of billions around the world. Efficient process management is a like trying to juggle knives while being on a unicycle. It's a complicated endeavor that involves many tasks such as task scheduling, memory allocation, and resource management. To give people a good experience with their phone you need to do all those things right. The way Android and iOS tackle this problem varies significantly. Each reflect the team behind them, their philosophies, and objectives., practitioners, and developers seeking to optimize system performance and stability in various computing environments. Android and iOS may both be mobile operating systems, but their approaches to maintaining system stability are about as different as night and day. iOS takes a rather simple approach by limiting access to system resources in order to keep the user experience consistent. This also gives them the benefit of a strong emphasis on security. That's not the case for Android, they choose to allow users to have more control over their device. Offering an open-source system for anyone who wants it. Of course this means you're more likely to run into issues caused by instability or malware. This paper closely examines how these two giants manage processes within their software, not just on what they do differently but also what similarities can be found.

## LITERATURE REVIEW

The prominence of mobile operating systems, exemplified by Android and iOS, has ushered in a new era of technological innovation, reshaping the landscape of smartphones and tablets. In this section, we delve into the extensive body of research surrounding process

management in Android and iOS. This literature review aims to provide an in-depth understanding of the evolution, challenges, and fundamental mechanisms that underlie these operating systems.

#### ANDROID PROCESS MANAGEMENT

Extensive research and development efforts have been dedicated to unraveling the intricacies of process management within the Android ecosystem. Android's task scheduling relies on the Linux-based kernel, which incorporates the Completely Fair Scheduler (CFS) to allocate processor time to processes [1]. Memory management is achieved through a blend of the Zygote process, the Dalvik Virtual Machine (DVM), and the more recent Android Runtime (ART) [2]. These mechanisms have witnessed continuous refinement over the years, with the transition from DVM to ART in Android 5.0 marking a substantial enhancement in memory management efficiency.

Resource management in Android involves processes and applications being allocated to CPU cores, ensuring the effective utilization of hardware resources. Recent Android iterations have introduced features such as the Background Task Manager to prioritize foreground processes and optimize power consumption [3]. Ongoing research in this realm centers on the development of resource-efficient task scheduling algorithms, memory optimization strategies, and power management techniques [4].

#### IOS PROCESS MANAGEMENT

Despite the proprietary nature of iOS, researchers have made notable strides in understanding and improving its process management mechanisms. iOS combines static and dynamic techniques to efficiently manage processes and their states, encompassing state preservation and restoration, background execution, and a secure sandboxing mechanism that isolates application processes [5]. Precise control over resource management and security stands as a hallmark of Apple's approach.

An area of particular interest in iOS process management research is energy efficiency. iOS introduced App Nap to conserve energy by slowing down apps running in the background. Research has focused on optimizing energy consumption while ensuring uninterrupted user experiences [6]. Additionally, investigations have delved into iOS's background execution, task scheduling, and memory management strategies [7].

#### PROCESS MANAGEMENT IN ANDROID

Android, an open-source mobile operating system, incorporates a sophisticated process management system that aims to deliver efficient and responsive performance across a diverse range of devices. Process management within Android encompasses pivotal functions, including task scheduling, memory allocation, and resource management. Here's an in-depth exploration of these core aspects:

#### TASK SCHEDULING:

**Linux Kernel:** Android relies on the Linux-based kernel for task scheduling. Through the utilization of the Completely Fair Scheduler (CFS) algorithm, the kernel allocates CPU time to different processes. CFS ensures fair distribution of CPU resources, preventing any one application from dominating the processor and thereby maintaining system responsiveness.

#### MEMORY MANAGEMENT:

**Zygote Process:** Android adopts a unique approach to memory management. Instead of creating a new process entirely from scratch when an Android application is launched, it leverages a Zygote process—a template containing essential system libraries and resources. New application processes are then created by forking from the Zygote process, significantly reducing memory overhead and expediting the process creation.

**Dalvik Virtual Machine (DVM) and Android Runtime (ART):** In earlier Android versions, like those predating Android 5.0, applications ran within the Dalvik Virtual Machine (DVM) using Just-In-Time (JIT) compilation. With the shift to Android 5.0 and beyond, Android transitioned to the Android Runtime (ART), employing Ahead-Of-Time (AOT) compilation. This transition markedly enhanced application performance and memory management by compiling code during installation, thereby reducing CPU and memory overhead and improving the overall user experience.

#### RESOURCE MANAGEMENT:

**CPU Core Allocation:** Android strategically allocates processes and applications to CPU cores, especially in multi-core devices, aiming to optimize hardware resource utilization. This allocation strategy prevents any single application from excessively monopolizing the CPU, promoting equitable resource distribution.

#### BACKGROUND TASK MANAGEMENT:

Recent Android iterations introduced features such as the Background Task Manager, prioritizing foreground processes over background tasks to optimize power consumption. Background tasks are categorized based on their importance, ensuring that critical tasks receive preferential treatment.

The process management system in Android has continually evolved to meet the demands of increasingly complex applications and diverse hardware configurations. The open ecosystem of the platform grants developers the flexibility to create a diverse range of innovative applications. However, this openness can sometimes lead to fragmentation, where different devices and Android versions may exhibit behavioral variations. Challenges in Android process management include optimizing energy consumption, developing efficient

task scheduling algorithms, and effectively managing system complexity. As mobile devices progress,

continuous research and development in these areas remain crucial to ensuring optimal performance and user experiences across Android devices.

In summary, process management serves as a cornerstone of Android's architecture, encompassing critical functions such as task scheduling, memory allocation, and resource management. These mechanisms have evolved over time, significantly impacting the performance and user satisfaction of Android devices. Researchers and developers persistently explore ways to enhance these processes and address the evolving challenges within the Android ecosystem.

#### PROCESS MANAGEMENT IN IOS

iOS, Apple's closed-source mobile operating system, incorporates a sophisticated process management system designed to ensure efficiency and responsiveness across its range of devices. Process management within iOS encompasses critical functions, including state preservation, background execution, and memory management. Here's an in-depth exploration of these core aspects:

##### STATE PRESERVATION AND RESTORATION:

iOS employs a combination of static and dynamic techniques to manage processes and their states effectively. State preservation and restoration allow applications to save their current state and later restore that state upon reopening. This ensures a seamless user experience, as users can continue where they left off within an application.

##### BACKGROUND EXECUTION:

iOS optimizes background execution by managing processes that run in the background. This mechanism allows certain types of tasks, such as music playback or location updates, to continue running even when the app is not actively in use. Apple has set specific criteria for applications to qualify for background execution, prioritizing essential functionalities.

##### MEMORY MANAGEMENT:

iOS maintains a secure sandboxing mechanism that isolates application processes, ensuring stability and security. The system manages memory allocation efficiently, preventing applications from accessing each other's memory space and maintaining a high level of security. This strict control over memory access contributes significantly to iOS' reputation for stability and reliability.

iOS's process management system is built on a foundation of strict control over resource management, emphasizing security, stability, and a consistent user experience. Apple's focus on curated experiences has led to a closed ecosystem that ensures a standardized and secure environment for users and developers. However, this control limits the customization of process

management and the freedom offered to developers in comparison to the open Android ecosystem. Challenges in iOS process management include balancing energy consumption with providing a seamless user experience, optimizing background execution while preserving battery life, and managing memory efficiently without compromising security and stability.

In summary, process management forms the core of iOS' architecture, incorporating functions such as state preservation, background execution, and memory management. These mechanisms have been meticulously designed to deliver a secure and consistent user experience, positioning iOS as a stable and reliable operating system. Ongoing research and development in these areas remain pivotal to maintaining and enhancing the high standards of performance and security within the iOS ecosystem.

#### SECURITY ISSUES IN IOS AND ANDROID: CYBER ATTACK VULNERABILITIES

Both iOS and Android face diverse security concerns, where cyber attacks during process management pose substantial risks to users. Android, owing to its open-source nature, presents increased vulnerability to cyber attacks. Malware infiltration and exploitation of vulnerabilities during task scheduling or memory management can compromise device security. Exploiting fragmentation, cybercriminals take advantage of delayed updates to target older Android versions with known security flaws. Similarly, iOS, despite its closed ecosystem and stringent app review process, is not immune to cyber attacks during process management. Exploitation of flaws in state preservation or background execution could result in malicious activities. Additionally, jailbroken or rooted devices, enabling customization but bypassing security measures, are prone to cyber intrusions, weakening inherent security protocols. These vulnerabilities within process management serve as entry points for cyber attacks, emphasizing the critical importance of timely updates, robust security protocols, and user awareness to mitigate potential threats across both iOS and Android platforms.

#### PERFORMANCE AND ANALYSIS WITH OPTIMIZATIONS

Optimizing process management in Android and iOS involves refining various aspects to enhance performance and efficiency. In Android, strategies focus on fine-tuning task scheduling algorithms for fair CPU time allocation and memory optimization to address fragmentation issues among diverse devices. Unified and timely software updates are pivotal to mitigate vulnerabilities arising from delayed updates, necessitating a more comprehensive and uniform update distribution. Advanced resource allocation algorithms are crucial to optimize resource usage across varying

hardware configurations. In iOS, optimization efforts center on advanced task handling through the Grand Central Dispatch and Mach Scheduler, efficient memory management leveraging the standardized ecosystem, and consistent software updates to maintain security. Refined resource management techniques adapt to evolving hardware while ensuring standardized and efficient resource allocation strategies. Continuous research and development in both platforms aim to minimize fragmentation, enhance system efficiency, and improve overall user experiences across Android and iOS devices.

## CHALLENGES AND FUTURE SCOPE

### Challenges:

- **Multitasking Efficiency:** Achieving a delicate balance between multitasking efficiency and power consumption poses a challenge. Optimizing background processes to operate seamlessly without unduly draining device resources is vital for a sustained user experience.
- **Real-time Performance:** Consistently delivering real-time performance remains a challenge. Ensuring immediate and predictable responses in diverse usage scenarios is an area requiring enhancement in both Android and iOS environments.

### FUTURE SCOPE:

- **Edge Computing Integration:** Exploring the integration of edge computing into process management can leverage localized processing near end-users, potentially reducing latency and improving system efficiency.
- **Machine Learning for Resource Prediction:** Implementing machine learning algorithms to predict resource needs and user behavior patterns can enhance resource allocation, optimizing device performance.
- **Enhanced Virtualization and Containerization:** Advancements in virtualization and containerization technologies offer improved isolation and control over processes, potentially elevating security and performance standards.
- **Collaborative Ecosystem Development:** Encouraging cohesive efforts among device manufacturers, developers, and OS providers to foster a more unified ecosystem can address fragmentation issues, ensuring standardized and expedited updates.
- **Quantum Computing and Future Architectures:** Investigating the potential implications of quantum computing and innovative architectures in redefining process management may unlock unparalleled performance and security capabilities.

## CONCLUSION

In summary, the examination of process management in Android and iOS reveals a landscape marked by multifaceted challenges and promising future prospects. Challenges, encompassing fragmentation, security

vulnerabilities, multitasking efficiency, and real-time performance, are evident in both ecosystems. Nevertheless, the future holds promise with potential advancements such as the integration of edge computing, the application of machine learning for predictive resource allocation, advancements in virtualization and containerization, collaborative ecosystem development, and the exploration of quantum computing and innovative architectures. Addressing these challenges and embracing forthcoming innovations has the potential to redefine process management, optimizing efficiency, security, and uniformity across Android and iOS. Through collaborative efforts among stakeholders and a commitment to cohesive ecosystem development, the horizon appears bright for the evolution of next-generation mobile operating systems, aiming to deliver enhanced user experiences and establish new industry standards.

## REFERENCES

- [1] Saman Paudel and Amul Neupane, CSC 725, Analysis of Scheduling Algorithm on the basis of battery consumption using an Android operating system, 2019.
- [2] Ruqiya and Noman Khan, Usability Analysis of Android and iOS Operating System, 2020
- [3] Prince Tehseen Ganai, Nisar Ahmad Malik and Khurshaed Ahmad Dar, Smartphone: Android Vs IOS, Vol. 1 October 2013.
- [4] Deepali Kayande and Urmila Shrawankar, Priority Based Pre-emptive Task Scheduling for Android Operating System, vol. 2, October 2011.
- [5] <https://ivypanda.com/essays/iphone-operating-systems-analysis/>
- [6] [https://www.researchgate.net/publication/335221195\\_Smartphone\\_Android\\_Vs\\_IOS](https://www.researchgate.net/publication/335221195_Smartphone_Android_Vs_IOS)
- [7] [https://www.researchgate.net/publication/261490936\\_Comparison\\_between\\_android\\_and\\_iOS\\_Operating\\_System\\_in\\_terms\\_of\\_security](https://www.researchgate.net/publication/261490936_Comparison_between_android_and_iOS_Operating_System_in_terms_of_security)
- [8] <https://www.frontiersin.org/articles/10.3389/fpsyg.2023.1040180/full>
- [9] [https://thesai.org/Downloads/Volume8No11/Paper\\_23-A\\_Comparative\\_Study\\_between\\_Applications\\_Developed.pdf](https://thesai.org/Downloads/Volume8No11/Paper_23-A_Comparative_Study_between_Applications_Developed.pdf)
- [10] [https://www.diffen.com/difference/Android\\_vs\\_iOS](https://www.diffen.com/difference/Android_vs_iOS)
- [11] <https://www.geeksforgeeks.org/difference-between-ios-and-android/>
- [12] <https://www.coursehero.com/file/p24garkm/Android-operating-system-uses-O-1-scheduling-algorithm-as-it-is-based-on-Linux/>
- [13] <https://arxiv.org/ftp/arxiv/papers/1304/1304.7889.pdf>