Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 14 Issue: 1s
Article Received: 29 January 2023 **Revised**: 24 March 2023 **Accepted**: 20 April 2023

Study of Kernels in Different Operating Systems in Mobile Devices

[1]Harshvardhan, [2]Shridhar Irabatti
[1]Research Scholar, MCA
Thakur Institute of Management Studies, Career Development & Research (TIMSCDR)
Mumbai, India
hvardhan223@gmail.com
[2]Research Scholar, MCA
Thakur Institute of Management Studies, Career Development & Research (TIMSCDR)
Mumbai, India
shridharirabatti22@gmail.com

Abstract— This Paper explains the three different types of kernel architectures that are used in different mobile operating systems namely - Monolithic kernel, MicroKernel and the hybrid kernel. The kernel is the core of any operating system, this document explains and differentiate the use of different kernels in different operating systems like Android, Windows mobile Operating system, Fuchsia, Ios etc. This analysis provides a clear understanding of kernels and its selections with respect to mobile devices. This end result shows that there is not a very significant difference in performance and optimizations in the operating system because of different kernel type selection.

Keywords— OS, Kernels, SOC, Monolithic kernel, Microkernel, Hybrid Kernel.

I. INTRODUCTION

Operating system Kernel is a key component of an operating system that manages operations of computer's hardware. It basically manages all the operations of memory and CPU. It is the core component of an operating system. Kernel acts as a bridge between applications and data processing performed at hardware level using inter-process communication and system calls

Kernels are loaded first into memory when an operating system is booted and remains into memory till the operating system is shut down again. It is responsible for various tasks such as disk management, task management, and memory management. There are different types of kernel and choice of kernel depends on specific requirements of the operating system and the goals of its developers.

II. LITERATURE REVIEW

G. Scott and K. Jeffay [16] in this paper, the authors propose a hybrid kernel design that combines the advantages of both monolithic and microkernel architectures. They argue that a hybrid kernel can provide a good balance between performance and flexibility, and can be used to build real-time and general-purpose operating systems that are both efficient and reliable. The paper also provides a detailed description of the design and implementation of a prototype hybrid kernel and evaluates its performance using a set of real-time and general-purpose benchmarks.

Tanenbaum [15] compares the design principles and implementation details of monolithic kernels and microkernels, and argues that monolithic kernels are more efficient and easier to implement than microkernels. He also presents a detailed analysis of the trade-offs between the two types of kernels, and shows how monolithic kernels can be used to build more efficient and reliable operating systems. The paper also compares the monolithic kernel and microkernel design with examples of popular operating systems like UNIX, Windows NT, and OS/2.

Sun et al. in [10] conducted an analysis of the relationships within the kernel of the Android operating system by creating a dynamic network model. Each node in the network represented a function and connections

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 14 Issue: 1s
Article Received: 29 January 2023 **Revised**: 24 March 2023 **Accepted**: 20 April 2023

represented the different call relationships between them. Through community research, they identified three distinct relationships between topological statistics and population size. Additionally, they performed a percolation study to identify fundamental mechanisms in software networks and to understand the organizational scale of the system. The results of this study may help to shed light on the complexities of the system and inform the development of appropriate methods for software testing.

Iqbal et al. [11] addressed the issue of security vulnerabilities in the Android mobile operating system, which is built on a simple Linux kernel and designed for use on touch-screen platforms such as tablets and smartphones. Due to a lack of adequate OS protections, the system is susceptible to a variety of security attacks that can restrict the access of third-party applications to sensitive infrastructure. They proposed an optimal permission elimination solution and demonstrated a way to prevent the exploitation of application authorization by implementing a shared User ID set for applications. They also presented a tool-based solution to prevent the breach of user information and ensure security against various types of Android protection threats and different permission types for Android applications.

Wang et al. in [12] introduced CrowdNet, a kernel-based architecture for cloud computing platforms that focuses on addressing Android kernel activity. The architecture features an automated data provider that gathers kernel footprints and a parallel malware forecast to validate malicious activity on Android. The authors used a heuristic approach on 12,750 Android apps to select hidden centers, reducing iteration and complexity. The results of their experiments indicate that CrowdNet effectively protected large-scale data validation and improved kernel learning, resulting in increased classification performance in detecting malicious attacks when compared to conventional neural networks and other machine learning methods.

III. TYPES OF KERNELS

A. Monolithic Kernel

Monolithic kernels are the most traditional type of kernel. They are a single, large block of code that contains all the core functions of the operating system, such as device drivers, system calls, and memory management. Because all of these functions are contained within the kernel, they can be easily accessed and executed by other parts of the system. This makes monolithic kernels simple and efficient, but it also makes them difficult to modify and extend. For example, if a bug is found in the device driver code, the entire kernel must be recompiled and re-installed to fix the bug. Operating systems with monolithic kernels such as OpenVMS, Linux, BSD, SunOS, AIX, and MULTICS can dynamically load (and unload) executable modules at runtime. This modularity is at binary (image) level and not at the architecture level. Modular monolithic operating systems are not to be confused with the architectural level of modularity inherent in server-client operating systems which use microkernels and servers (not to be mistaken for modules or daemons).

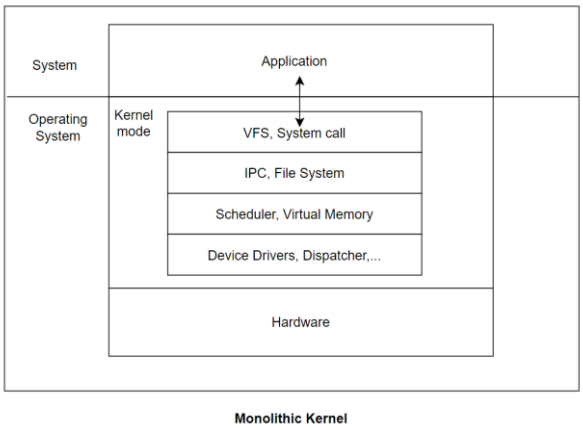

Fig. 1. Monolithic Kernel

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 14 Issue: 1s
Article Received: 29 January 2023 **Revised**: 24 March 2023 **Accepted**: 20 April 2023

Examples

1. Android

Most famous Mobile operating System Android uses Linux kernel which is a monolithic architecture and the market share is more than 2.5 billion devices currently.

Google has made a number of architectural changes to the Linux kernel in its implementation of Android, such as the addition of device trees, ashmem, ION, and unique out-of-memory handling. These changes were made outside of the typical Linux kernel development process. Some of the features that Google has contributed back to the Linux kernel, such as the "wakelocks" power management feature, have been rejected by Linux kernel developers, partly due to concerns about maintenance. In 2010, Google announced plans to hire employees to work with the Linux kernel community, but the current Linux kernel maintainer has expressed concerns that Google is no longer trying to get its code changes included in the mainstream Linux kernel. Google has stated that "Android is not Linux" and it has very little in common with the conventional desktop Linux stack, and the benchmark score of the latest Android device is 269627 in ANTUTU for CPU and 251098 for Memory.

2. Ubuntu touch

Ubuntu Touch is a mobile version of the Ubuntu operating system developed by the UBports community. It is designed to run on smartphones and tablets, and it uses the same codebase as the desktop version of Ubuntu. Ubuntu Touch features a touch-friendly interface and is optimized for use on mobile devices, with support for gestures, notifications, and other mobile-specific features. It also includes a variety of apps and services, such as the Ubuntu Store, which allows users to download and install apps and games. Ubuntu Touch is available for a number of different devices, including the Nexus 4, Nexus 5, and Nexus 7.

B. Microkernel

A microkernel is a type of operating system (OS) kernel that contains only the essential core components necessary to provide basic services to other system components. These core components, such as memory management and process management, are typically implemented as system calls that can be accessed by other components through a well-defined interface.

The microkernel architecture is designed to make the OS more modular, flexible, and secure. By separating the kernel into small, isolated components, it is possible to add or remove functionality as needed, and to update individual components without affecting the entire system. This also makes the kernel more robust, as bugs or security vulnerabilities in one component will not necessarily affect the entire system.

One of the main advantages of a microkernel-based OS is its ability to support different types of hardware and software. By providing a minimal set of core services, the microkernel allows other system components to be implemented as user-level modules, which can be easily replaced or updated without affecting the rest of the system. This makes it easier to add new hardware or software support, and to adapt the OS to different types of devices and platforms.

Examples of microkernel-based operating systems include QNX, L4, and the Hurd of GNU.

However, it's worth noting that microkernels can have some drawbacks as well. They can have higher overhead due to the need to make system calls, which can slow down system performance. Additionally, they can be more difficult to develop and maintain, as it can be challenging to ensure that the various components of the system work together seamlessly.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 14 Issue: 1s
Article Received: 29 January 2023 **Revised**: 24 March 2023 **Accepted**: 20 April 2023

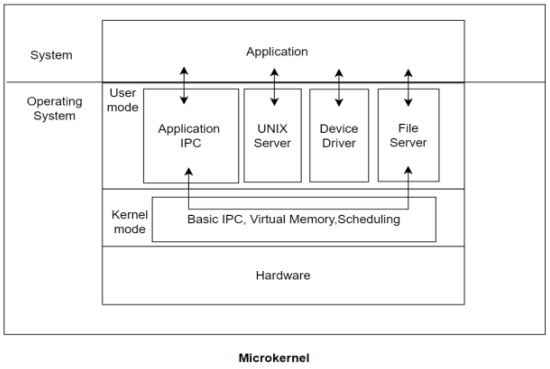

Fig. 2. Microkernel

Examples

1. Fuchsia

Fuchsia is an operating system being developed by Google. It is not based on any existing operating system, such as Linux or Windows, and is designed to run on a wide range of devices, including smartphones, tablets, laptops, and smart home devices.

Fuchsia uses a microkernel called "Zircon" and its user interface is based on Google's Material Design guidelines. It also includes a new programming language called "Flutter" that is used to build the system's user interface.

One of the unique features of Fuchsia is its use of a new type of application architecture called "Flutter apps", which allows for more flexibility and scalability in terms of how applications are built and run on the system. Additionally, Fuchsia is designed to be highly secure and to support multiple user accounts.

It's worth noting that Fuchsia is still in the early stages of development and it's not clear when or if it will be released to the public. Google has not announced any official plans for the operating system and it's not clear what the company's long-term goals are for it.

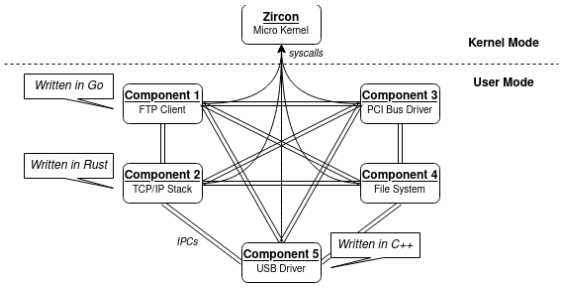

Fig. 3. Fuchsia

C. Hybrid Kernel

A hybrid kernel is a type of operating system (OS) kernel that combines features of both microkernels and monolithic kernels. A microkernel only includes the essential core components necessary to provide basic services to other system components. While a monolithic kernel has all of the kernel code in one large block, including device drivers and file system.

A hybrid kernel, as the name suggests, combines elements of both microkernels and monolithic kernels. It includes a small microkernel core that provides basic services such as memory management and process management, as well as a set of device drivers and file systems that run in user space as separate processes. This

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 14 Issue: 1s
Article Received: 29 January 2023 **Revised**: 24 March 2023 **Accepted**: 20 April 2023

means that the hybrid kernel can take advantage of the modularity and flexibility of a microkernel, while also providing the performance benefits of a monolithic kernel.

The main advantage of a hybrid kernel is that it can provide a balance between the security and flexibility of a microkernel, and the performance and ease of development of a monolithic kernel.

This makes it well suited for use in systems that require a balance of performance and security, such as embedded systems or mobile devices.

Examples of hybrid kernel-based operating systems include Apple's iOS, and the Windows NT kernel used by Windows NT, 2000, XP, Vista, 7, 8, 8.1 and 10.

It's worth noting that the distinction between microkernel, monolithic kernel and hybrid kernel can be somewhat blurry, as some operating systems that are classified as one type may have features of another type. And some operating systems are considered hybrid but are not following the exact definition of it.

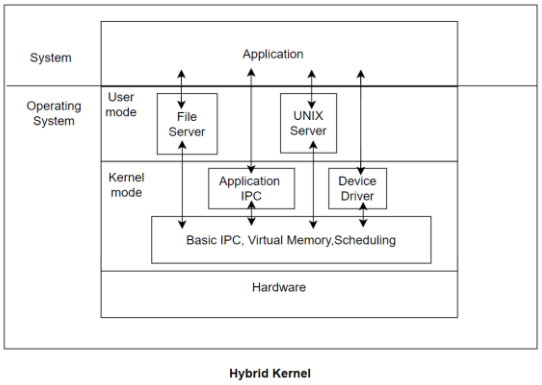

Fig. 4. Hybrid Kernel

Examples

1. IOS

The kernel of Apple's iOS operating system is based on the open-source Darwin operating system, which is built on top of the XNU kernel. The XNU kernel is a hybrid kernel, which means it combines features of both microkernel and monolithic kernels. It includes features such as memory management, task scheduling, and support for multiple architectures. The iOS kernel provides the foundation for the rest of the operating system, including the user interface and the various apps that run on it. The latest version of IOS scored a benchmark of 247747 in ANTUTU which is lower than the competitor Android.

2. Windows Mobile

Windows Mobile used the Windows CE kernel, which is a compact version of the Windows NT kernel. The Windows CE kernel is a hybrid kernel, which means it combines features of both microkernel and monolithic kernels. It includes features such as memory management, task scheduling, and support for multiple architectures. The Windows Mobile operating system provided the foundation for the user interface and the various apps that ran on it.

Windows mobile was discontinued in 2010 and replaced by Windows Phone.

IV. CONCLUSION

The choice of kernel architecture depends on the specific requirements of the operating system and the goals of its developers. Each type of kernel has its own advantages and disadvantages, and the best choice will depend on the specific use case.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 14 Issue: 1s
Article Received: 29 January 2023 **Revised**: 24 March 2023 **Accepted**: 20 April 2023

Microkernel architecture is best suited for systems that require high levels of security, modularity, and ease of development. This is because the architecture separates the operating system services into different user-mode processes, which makes it easier to isolate and secure individual components. Also, it allows for more flexible development and maintenance, as individual components can be developed and updated independently of each other.

Monolithic kernels are best suited for systems that require high performance and efficient use of system resources. Because all the operating system services run in kernel mode, they have direct access to system resources, which allows for faster and more efficient communication between components.

Hybrid kernel architecture is a compromise between the Microkernel and Monolithic architecture, it tries to combine the best of both worlds and provide a balance of security, modularity, and performance. It's best for systems that require a balance of security, modularity, and performance.

Ultimately, the choice of kernel architecture will depend on the specific requirements of the operating system and the goals of its developers. Some popular operating systems like Linux, Windows, and MacOS use Monolithic kernel, while others like Fuchsia, L4 use Microkernel, and some like IOS,Windows Phone, osDarwin, use Hybrid kernel.

The benchmark of score of different OS depends on various factors like the CPU used and the architecture of it, currently most of the mobile devices SOC is based on ARM architecture. Apart from that the number of cores in the SOC also play a crucial role, recently most vendors of SOC like MediaTek and Qualcomm used different architectures for different cores in the SOC to manage the Power consumption and Performance. Mostly there are two divisions in the cores one is Performance core for heavy computing and the other is efficiency core for better battery life of the devices. So, the benchmark scores can’t be used to select the kernel type of the OS.

References

[1] https://en.wikipedia.org/wiki/Hybrid_kernel
[2] https://en.wikipedia.org/wiki/Monolithic_kernel
[3] https://en.wikipedia.org/wiki/Microkernel
[4] https://en.wikipedia.org/wiki/Windows_Mobile
[5] https://en.wikipedia.org/wiki/Ubuntu_Touch
[6] https://en.wikipedia.org/wiki/Fuchsia_(operating_system)
[7] https://en.wikipedia.org/wiki/Android
[8] https://en.wikipedia.org/wiki/IOS
[9] https://www.geeksforgeeks.org/kernel-in-operating-system/
[10] Function-call network reliability of kernel in android operating system by Sun P, Yang S, Lai Z, Li D, Yao A. in IEEE 2019.
[11] Android (Nougats) security issues and solutions by Iqbal S, Yasin A, Naqash T. in 2018 IEEE .
[12] Wang X, Li C, Song D. CrowdNet:identifying large-scale malicious attacks over android kernel structures in IEEE 2020.
[13] Kim J, Shin P, Kim M, Hong S. MemoryAware fair-share scheduling for improved performance isolation in the Linux kernel. IEEE Access
[14] Bala K, Sharma S, Kaur G. A study on smartphone based operating system.International Journal of Computer Applications 2015.
[15] Monolithic Kernels vs Microkernels" by Andrew S. Tanenbaum, published in the Proceedings of the IEEE in 1992.
[16] Hybrid Kernel Design for Real-Time and General-Purpose Systems" by G. Scott and K. Jeffay, published in the Proceedings of the Real-Time Systems Symposium in 1999.