

Differences between Embedded System Operating Systems and Mobile Operating Systems

Wesly Yii

BSc (Hons) in Computer Science
Sunway University
Selangor, Malaysia
18066233@imail.sunway.edu.my

Tan Chun Ze

Bachelor of Software Engineering (Hons)
Sunway University
Selangor, Malaysia
15020605@imail.sunway.edu.my

Lee Ming Zhen

Bachelor of Software Engineering (Hons)
Sunway University
Selangor, Malaysia
18088757@imail.sunway.edu.my

Ryan Aung Yi Yang

BSc (Hons) Computer Networking and Security
Sunway University
Selangor, Malaysia
18007906@imail.sunway.edu.my

Abstract—Operating systems play a significant role in powering devices worldwide, this document seeks to provide a comparison between Embedded system operating systems and Mobile Operating systems.

Keywords—embedded systems, mobile operating systems, operating systems, difference, compare

I. INTRODUCTION

Human activity is increasingly augmented with computing systems, and the hardware they run on usually have some kind of operating system on it, be it the phones in our pockets that use a Mobile Operating System, or the Subway ticket door that uses an Embedded system operating system.

In this document, we seek to provide a glimpse into the world of these two types of operating systems, by talking about each of their relevance, usage scenarios, pros and cons. We will also provide a brief comparison between Microkernel and monolithic kernel.

II. BACKGROUND

A. Embedded System Operating Systems

a) What is it about?:

Embedded system operating system is a type of operating system that carries out distinct processes for particular devices in order for them to operate optimally. It is just like a toned down version of a standard computer OS with limited functionality [1]. Embedded system operating systems may exist as a standalone independent system or belong as a part of something bigger. There are five types of embedded operating system; Multi-Tasking Operating System, Rate Monotonic Operating System, Preemptive Operating System, Single System Control Loop and Real Time Operating System. Generally, Its main focus is to let a certain device function as intended by running its codes [2].

b) Why the need for it?:

Mobile OS is originally derived from the computer OS which was then evolved into an embedded OS and then to the present smartphone-oriented OS. Throughout the evolutions, the architecture of mobile OS has become simpler compared to the previous versions. In terms of hardware, the factor size of microprocessors and peripherals has been reduced to accommodate the modern design of smartphones.

Technically, smartphones do not require an operating system and the device can just run without problem but it would be a mess since there will be no operating system present to manage the memory, storage and applications. Therefore, the main reason for embedded OS to exist is to act as a partitioning tool as well as manage hardware and software resources for the embedded system to work as intended. It also simplifies software development in the upper layer by providing an abstraction layer. The need for it lies in how embedded operating systems can perform a dedicated function for a specific device to run while also allow it to be less demanding in terms of resource consumption [3].

c) What is the application?:

Applications of embedded system OS include Embedded Linux, iOS, Windows Mobile OS, Blackberry OS, and Symbian. A Linux operating system that is utilized within embedded devices and appliances is called Embedded Linux. ATM machines utilising Windows XP is also an application of embedded operating systems.

B. Mobile Operating Systems

a) What is it about?:

A mobile OS is another type of operating system that is created for the sole purpose of powering mobile devices. These mobile devices include PDAs, tablets, smartphones etc. Similar to how the Windows operating system works in a desktop, a mobile operating system allows different

kinds of applications to run on a mobile device by serving as a software platform beneath the apps. All the features and functions that are available on mobile devices are all set by the corresponding mobile operating system. On top of that, mobile operating systems will also be in control of the third-party programs that are used on the mobile device.

b) Why the need for it?:

A mobile operating system is essential for mobile devices to run ideally. A mobile operating system would enable multitasking, which means that multiple applications can run at one time. It will also allow the users to adjust the brightness and volume of all the applications through a single button. Other than that, applications would then be compatible and portable across multiple devices of different brands.

c) What is the application?:

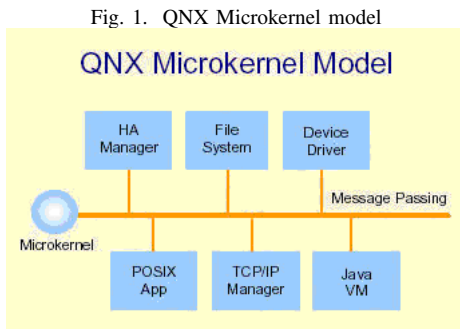
Some notable examples of mobile OS are Apples iOS and Googles Android OS. The differences between iOS and Android is that iOS is designed to only run on Apple devices within its XNU kernel while Android is an open source software which implies that it can be used by other mobile device manufacturers who are then able to customize the Android source code to fit their own devices needs. Apple has also released other mobile OS meant for their Apple watch and iPad which is watchOS and iPadOS respectively [4].

Covering all types of operating systems in embedded systems and mobile systems can be vague and challenging. With that said, Operating systems that are ubiquitous will be chosen to elaborate more in depth about their structure and features.

III. IN-DEPTH REVIEW AND DISCUSSION

a) Microkernel in QNX Architecture:

QNX is one of the operating systems used in real-time embedded systems. QNX heavily relies on Microkernel architecture to support its own operating system [5] [6].



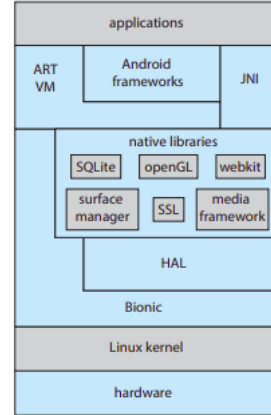
Microkernel only provides fundamental OS services including interprocess communication(IPC), memory management and CPU scheduling [6]. Other services are implemented as user-level programs located in separate address space from Microkernel [6]. To achieve extensible service implementation

and communication between other services and microkernels. IPC is an important function as it uses message passing [5] allowing other services to exchange messages between each other indirectly.

b) Linux kernel in Android architecture:

Android OS has a 86.1% market share in the mobile OS market [7]. It is based on the Linux kernel with a monolithic structure.

Fig. 2. Android architecture



Fundamental OS services and other non-essential services are combined together into one single address space [6]. System calls are used in android architect to allow application programs to interact with the kernel [8].

Figure 2 indicates that androids architecture consists of a layered stack of software providing an abundance set of frameworks and libraries to support crucial hardware features including graphics, audio [6]. For that reason, it gives the android operating system to be compatible with any types of android enabled devices.

With the accumulated resources gathered, it can be concluded that embedded OS and mobile OS use different approaches on OS structure for different benefits, and using the wrong OS type could be devastating. Thus, the pros and cons of embedded operating system and mobile operating system shall be explained to aid the choosing of OS architectures.

A. Pros of Embedded OS

a) Loads faster and consumes less power:

Embedded system OS can be attached to a PC and will always load faster from the flash compared to a standard computer OS because of the limited features built in it. This is beneficial when users only need to perform a single task such as browsing the internet [8].

b) Distinct functions:

Embedded system OS differs greatly from standard OS in terms of their abilities to perform different functions. Embedded OS will be limited to one single task and will perform the task regardless of user intervention. Examples of

devices using an embedded OS would range from a simple toaster to specific embedded systems such as the QNX4 RIOS [9].

B. Cons of Embedded OS

a) Hardly Scalable:

Embedded system OS are usually not upgradable since its system stays fixed after the initial configuration. Some embedded systems OS like the ones in ATM machines are also designed to not receive upgrades to prevent tampering. Embedded system OS only allows upgrades if the chip is flashable. An example of an upgradable embedded system OS would be the one in a wifi router, it can be updated by flashing the card in the router upon downloading the new firmware [10].

C. Pros of Mobile OS

a) Open source platforms are constantly updated and well-supported:

Mobile OS such as Android open source software is one of the most widely used Mobile OS in the world. It is utilised by top mobile conglomerates like Huawei and Samsung. In recent years, we could see that there are more variations of the open source platform to support and cater to specific markets like the HongMengOS (Harmony OS) that is launched in 2019 for new Huawei devices exclusively [11]. Such new launches suggest that Mobile OS are still in high demand and will be the focal point of operating systems in many years to come.

b) Convenience and flexibility:

Mobile OS is a newer concept and is built to focus on being convenient to its users. It is improved on existing OS and opts to strive in responsive design, consistent and stable network access and possible improvements to be used in the diverse wireless environment [12].

Mobile OSs like Android are open-source and are widely used as a base for many relatable OS like HarmonyOS. The fact that it is open source opens itself to many possibilities. It is flexible because many others improve and adjust the software to cater to their own needs.

c) Scalability and maintainability:

Mobile OS outcompetes embedded OS in this regard by a wide margin. They are built to handle any number of connections from the user side, handling errors and backend API interactions [13]. The maintainability instead is a way to iterate and improve upon existing software and OS efficiently when it comes to debugging and further improvements.

D. Cons of Mobile OS

a) Instability:

Mobile OS struggles with the constraints of handling constant cellular interruptions and communications, power issues and the load on handling multiple applications launched by the user [14]. On average, a mobile phone easily uses anywhere

from 50 to 500 processes on a comparatively smaller capacity compared to an embedded system OS which commonly runs a single shell. This leads to frequent crashes and bugs that leads to complications and instabilities.

b) Battery / Power:

Mobile OS runs on a limited battery capacity unlike embedded OS which runs with a constant power supply. Although modern mobile devices now are equipped with better batteries, they are still lacking and unable to keep up with improvements in graphics and animations like in triple A games for mobiles. Many background processes are also capable of draining the battery, accelerating its wear.

IV. CONCLUSION

After looking into both embedded system OS and mobile OS, we can see that both types of OS have their own pros and cons. However, in terms of efficiency in performing what they are meant to do, QNX microkernel architecture is by far the most superior. Being small in size, it provides complete memory protection as it runs in the kernel space while other processes run in the user space. Most importantly, we think QNX is a good fit here as it is the best for mission critical applications due to the fact it never crashes thus making it robust, as any third party crashes does not impact the kernel itself and it keeps running at all times [15].

REFERENCES

- [1] T. Noergaard. Embedded Operating System. [Online]. Available: <https://www.sciencedirect.com/topics/computer-science/embedded-operating-system>
- [2] (2017, June) Introduction to Embedded Operating Systems. [Online]. Available: <https://microcontrollerslab.com/embedded-operating-system/>
- [3] P. Viswanathan. (2020, April) Introduction to Embedded Operating Systems. [Online]. Available: <https://www.lifewire.com/what-is-a-mobile-operating-system-2373340>
- [4] C. Steele. mobile operating system. [Online]. Available: <https://searchmobilecomputing.techtarget.com/definition/mobile-operating-system>
- [5] T. W. Burger. (2009, March) Embedded development - qnx* or linux*? [Online]. Available: <https://software.intel.com/content/www/us/en/develop/articles/embedded-development-qnx-or-linux.html>
- [6] P. B. G. Abraham Silberschatz, Greg Gagne, *Operating System Concepts, 10th Edition*. Wiley, May 2018.
- [7] I. D. Corporation. (2020, April) Smartphone market share. [Online]. Available: <https://www.idc.com/promo/smartphone-market-share/os>
- [8] T. differences. (2016, December) Difference between microkernel and monolithic kernel. [Online]. Available: <https://techdifferences.com/difference-between-microkernel-and-monolithic-kernel.html#KeyDifferences>
- [9] G. Jackson. Network operating systems vs. embedded operating systems. [Online]. Available: <https://smallbusiness.chron.com/network-operating-systems-vs-embedded-operating-systems-46508.html>
- [10] M. Kyrmin. (2020, January) Embedded operating systems on traditional pcs. [Online]. Available: <https://www.lifewire.com/embedded-operating-systems-833068>
- [11] Huawei. (2019, August) Huawei launches new distributed operating system, harmonyos. [Online]. Available: <https://consumer.huawei.com/my/press/news/2019/huawei-launches-harmonyos/>
- [12] Technopedia. What is the difference between a mobile os and a computer os? [Online]. Available: <https://www.techopedia.com/7/29694/software/what-is-the-difference-between-a-mobile-os-and-a-computer-os>
- [13] Y. Kobets. (2017, December) Scalability and maintainability of mobile and other applications. part 1. [Online]. Available: <https://medium.com/@yuri.live/scalability-and-maintainability-of-mobile-and-other-applications-part-1-21e3d07daded>

- [14] S. Litchfield. (2017, November) Every mobile os is on the verge of a breakdown and needs tlc. [Online]. Available: {http://allaboutwindowsphone.com/features/item/22588_Every_mobile_OS_is_on_the_verg.php}
- [15] S. N. T. (2017, June) What is the advantage of using qnx in embedded systems? [Online]. Available: <https://qr.ae/pNylkw>