Resource Management and Security issues in Mobile Phone Operating Systems: A Comparative Analysis

Khalid M. Awan[1], Muhammad Waqar[1], Muhammad Faseeh[1],Farman Ullah[2] and Muhammad Qaiser Saleem[3]

[1] Department of Computer Science, COMSATS Institute of Information Technology Attock, Pakistan
[2]Department of Electrical Engineering, COMSATS Institute of Information Technology Attock, Pakistan
[3]College of Computer Science and Information Technology, Al Baha University, Al Baha, Saudi Arabia.
[e-mail:, waqar2193@gmail.com, ifaseeh1@gmail.com, farmankttk@ciit-attock.edu.pk, muhammad.qaiser.saleem@gmail.com]

Khalid.awan@ciit-attock.edu.pk
*Corresponding author: Khalid Awan

Abstract

Nowadays Mobile phones are becoming more popular in our daily lives. Mobile technology has a great effect on human life. Our daily tasks are dependent on mobile devices. Memory Management (MM), Security and Performance plays an important role in every handheld device specially in mobile phones, which are very much dependent on their operating system (OS). These embedded operating systems are on the driving seat when we talk about efficient and useful memory management and secure handling. Three popular OS in mobile phones are Android, Windows and iOS (iPhone OS). Each OS has its own way of managing the memory and provide it to certain number of applications. Android is an open software available for the people to modify as per their needs. But Windows and iOS operating systems didn't allow their software as open source. Researchers have done a large amount of work using different mechanisms and decision makings to develop new ways to manage the memory of these OS's. This work shows a comparative analysis of different memory management and security related techniques in above three operating systems. In this paper, we present the analysis of memory management and security in mobile phone operating systems with respect to apps, main memory, cache memory and virtual memory. Also, we compare the overall performance of these OS's in terms of MM, security concerns. This study will help in finding better operating system in terms of efficient memory management and security.

Keywords: Memory Management, Security, Mobile Phones, Android, Windows OS, iOS (iPhone OS)

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3344v1 | CC BY 4.0 Open Access | rec: 13 Oct 2017, publ: 13 Oct 2017

1. Introduction

Mobile phone is becoming very popular device and is playing a vital role in our daily lives. As we know that number of mobile phone users are increasing day by day, most of our tasks are dependent on it. If we analyze our environment, even a poor person has a mobile phone. Mobile phones are coming with the number of different useful features like call features, calculations, maps etc. Most of the people are using the mobile phones to perform business tasks. Sending money, receiving money, paying bills and many other tasks are now become easier due to this technology.

Nowadays, smart phones are gradually becoming more faster just like a mini computer [1]. Using the smart phones for sending or receiving the emails, voice or text messages, browsing over the internet etc. most of the smart phones are also being used for taking photos, and making video etc. Due to all these functions, different mobile manufacturing companies are manufacturing the cell phones for different categories of people. Due to increase in competition, mobile manufacturing companies are increasing their mobile phone quality as well as decreasing the cost [2]. They are also providing the unique features for increasing the productivity. Apple, Huawei, Google, Samsung, Blackberry are the famous companies which are making the mobile phones.

In section 1.1 we discuss the different Mobile phones operating systems. Section 1.2, 1.3 and 1.4 elaborates the memory management, security and performance in mobile phones respectively.

1.1 Mobile Phone Operating Systems

In the race of mobile phones, another important and the core component is being missed. With the increasing competitions among the mobile manufacturers, Operating system of the mobile phone is also an important part. There are various operating systems for mobile phone is available in the market like Android OS which is developed by Google, Apple developed iOS, Windows OS is developed by Microsoft, and blackberry OS etc. [3].

With the increase of mobile phone quality, quality of mobile operating system is also important. Most of mobile phone users are worry about, which OS is better for them. Different mobile phone OS has different unique features. To resolve this issue, we are comparing the different features of three major OS in mobile phones, which include Memory Management, Security level and overall performance.

1.2 Memory Management

Memory is always a limited part especially when we talk about the handheld devices (Mobiles, Tablets etc.). Each Mobile phone uses some operating system which controls them according to the need. Nowadays, three largely used OS in mobile phones are Android, Windows, and iOS; in which android is the most common OS which are used by many companies specially Google [4]. Windows OS is specifically used in Microsoft devices while iOS (iPhone OS) is used by Apple in their devices. These three OS are always competing in market to dominate from the others so, the developers of these OS are always trying to develop new techniques [5].

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3344v1 | CC BY 4.0 Open Access | rec: 13 Oct 2017, publ: 13 Oct 2017

Memory management is a big issue when it comes to mobile phones. Every user wants that all the apps on their device should run smooth and fast. It is now the duty of mobile operating system to make sure that the user does not experience any inconvenience. For this purpose, many researchers have done a lot of study and applied different techniques to improve the performance of these OS with respect to MM. Generally, there are some systems apps which should be running all the time and thus should remain in memory forever. On the other hand, some of the apps which are used according to the need, when launched take some time to load into the memory and when the user is done with it, the app should either be terminated or not depending upon the memory. Now the time used by the app to load into the memory and the time when the user switches to another app which is already loaded in memory are the main issues to focus because both the times are directly depending upon the memory [6].

As there are some rules for memory management to follow for the best experience are: if no object is needed, release the object. Hold the ownership of an object that you didn't create until you must release these objects too when they are not needed. Last, no need to release the objects that is not under your ownership.

Existing research on android, windows and iOS tend to focus on memory management with respect to the app launch time and switching between two apps. They also focus on the fact about how the application crashes and what are the reasons behind it. Mobile OS has many type of memories like cache memory, RAM, virtual memory, internal memory and external memory. This paper also depicts a brief review on how these operating systems in mobile phones manage the memory when large number of apps are running in different scenarios. This paper also analyzes memory management in perspective of application launching time and switching between different applications, main memory, cache memory and virtual memory. A comparative analysis of given operating systems is also done to find out which operating system is best. Furthermore, several other features of these operating systems w.r.t MM is also discussed.

1.3 Security

In terms of security features, different OS's have different mechanisms for securing the mobile phones. There are different types of security concerns present today [7]. These problems are occurring in the operating systems due to less attention when designing the operating system of mobile device. Some of the people find out the loopholes, and attack on the mobile devices for different purposes. Some attacks for the stealing information, network information, personal data, logins (saved in mobile) etc. There are some of the applications which are designed for specific type of attack on operating systems of handheld devices. Usually these are third party applications. Sometimes, these applications change the system files or systems default settings which is playing a vital role in our mobile phone security [8]. So, in this paper, we will also compare the security concerns of different operating systems in mobile phones.

1.4 Performance

Performance is one of the core components of any device as well as of any operating system. Performance includes multiple parameters like execution speed of some task, quality of loading and running the apps, speed of switching between the different application running in the memory [9]. This paper also focuses on performance analysis of given mobile phone operating systems. Performance also include better time, better consumption level of battery etc., but we will only compare the execution speed in different operating systems of mobile phones.

…

In section 2, 3 and 4 we detail the techniques used in memory management and security for Android, iOS and Windows operating systems respectively. Section 5, 6, and 7 discusses the comparison between android and windows, android and iOS, and windows and iOS MM and security techniques with performance analysis respectively. At the end section 8 deliberates comparative analysis of all the three OS's android, windows and iOS. The final conclusions are discussed in section 9.

2. Android Operating System

Android operating system is Linux based, specially designed for the mobile phone as well as the tablet devices. First public release of the android operating system was launched in 12th November 2007. First mobile phone with Android operating system was launched in market on 23rd September 2008. Android is an open source operating system, based on Linux kernel. Linux kernel enable the developer to write or modify application which are initially present in Java. Linux Kernel also supports C/C++. Android provide easily access to public sites like YouTube, Facebook, Gmail, google calendar etc. Android is most popular software among the manufacturer because it is an open source operating system.

In section 2.1 we discuss the architecture of android OS, while section 2.2, 2.3 and 2.4 briefly elaborates the techniques used in memory management, security concerns and performance analysis of android OS respectively.

2.1 Android OS Architecture

Android OS is basically divided into four layers. Linux Kernel, Libraries, Application framework, and Applications revealed in **Fig. 2.1**. Linux kernel is responsible to provide virtual memory, drivers, power management and networking Local libraries layer provides the support of different software and built-in libraries like SQLite, SGL (primary 2D graphics provider). Web Kit is used for small screens. Application frame work provides API's, in which sharing and receiving data is being handled.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3344v1 | CC BY 4.0 Open Access | rec: 13 Oct 2017, publ: 13 Oct 2017

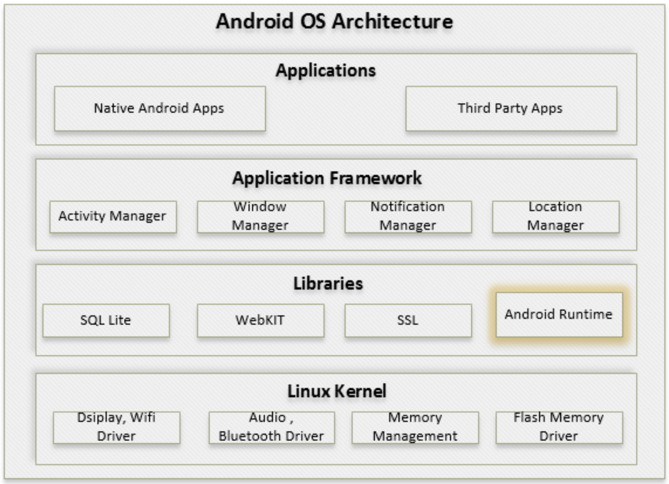

Fig. 2.1. Android OS Architecture

2.2 Memory Management in Android

Application launch time in android can affect the user badly and lot of software's are developed to reduce this time but not much work is done in hardware perspective. Kim et al. [10] proposed several hardware optimizations according to the application usage pattern and show that Nonvolatile memory (NVM) can speed up the start time of an application. In general android has a process called "Zygote" that provides share memory space for different application to share the same libraries, reducing the need to load the repeated libraries. Also, android keep an application in memory even after the application is ended to reduce the startup time in future. Most of the time DRAM is not used completely so dynamically adjusting its size can reduce power consumption of DRAM. Secondly NVM is used between flash and phase change memory (PCM) as secondary memory, and a portion of NVM is allocated to an application whenever it starts for the first time. PCM stores frequently used applications and their libraries. Experimental results show that it improves the performance of application start time but on the other hand it also reduces the memory space.

Android has a built-in system for memory management which is called "low memory killer". When memory is insufficient for new applications then android terminates processes until sufficient memory is available. The process to be terminated depend upon the predefined priority and memory consumption but this termination is sometime inconvenient for the user. Nomura et al. [11] proposed new policies on which processes should be terminated. First is LRU (least recently used) and second is application re-launching time. Experiment results proved to improve application launching time.

Memory management in android is done through Activity Manager Service (AMS) and Low Memory Killer (LMK) shown in **Fig. 2.2**. We have talked about how android kills process using LRU approach, but this may sometime kill those processes which are required in short interval of time. So, Vimal et al. [12] proposed a new technique of memory management that takes into account application usage pattern and decide accordingly while adjust cache management dynamically based on user's interest.

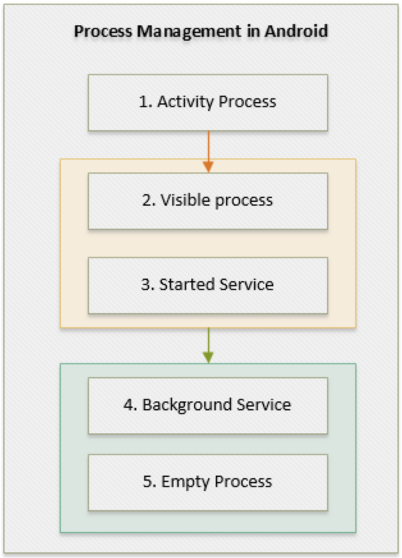

Fig. 2.2. Memory Management in Android

The internal memory of any mobile is limited which creates problem for a user who wants to have large number of apps installed at a time. Singh et al. [13] proposed a new technique for extending internal memory by using cloud storage, in which those apps which are unused or used often by the user are moved to the cloud until required by the user, while making internal memory available for new apps installation. The apps on cloud avoid permanent deletion and maintain user data.

Generally, applications running on android OS work smoothly but sometimes application crashes in the middle of its lifecycle. This happens due to the memory leaks in the system. As memory leaks are not easily identified, [14] a new technique is introduced to detect the vulnerabilities of memory leaks during application developing and testing by identifying certain of code patterns in applications. This allows the developers to explicitly detect the code in the applications which can cause the memory leaks.

2.3 Security in Android

Dalvik Virtual Machine (DVM) is responsible for handling the security in Android operating system [15]. In the main design of DVM, no application has guaranteed to perform any task that would dangerous for any other application, operating system and user. This technique is forcefully implemented using DVM and fundamental Linux platform using the file permission and UNIX user identifiers. Unlikely, many Linux Desktop OS's, where applications from the same user execute with the same user ID, and each application executes its own virtual machine in a different process with its own user ID. This clearly indicates that Android applications cannot access the data or code from other.

On the other side, Android is an open source software which is available for different manufacturers and organizations. Due to its open availability, there are many chances for different attacks on this operating system [16].

2.4 Performance in Android

Android OS is strongly focusing on the performance. As we know that Android is an open platform and many applications are developed publicly. For this purpose, Android always suggest two techniques for their developers which is also the best rule for the development process [17]. First, don’t do work that you don't need to do. Second, don’t allocate memory if you can avoid it.

For improving the performance, android is trying to utilize the less memory with large tasks. They are also working on reducing the processes, running in background. Improving the apps performance, assign the high priority to top running app.

3. iOS Operating System

Apple launched its first iPhone device on 29 June 2007 in USA. iPhone runs their own developed operating system called iOS built in Objective-C programming language [18] [19]. There are many other versions available with different iPhone devices. But the most current version is iOS 10 with most advanced and efficient features.

In section 3.1 we discuss the architecture of iOS, while section 3.2, 3.3 and 3.4 briefly elaborates the techniques used in memory management, security concerns and performance analysis of iOS respectively.

3.1 iOS Architecture

iOS architecture is basically consisting on 4 layers [20]. Core OS layer is available as the bottom layer. Core OS layers contain additional abstraction layer, called cocoa and touch layer. Core OS contains the scheduler, file system, Mach kernel, hardware drivers and in-charge of the memory system, network and Inter process communication, security framework to secure the system and program data. Core services layer contains abstraction setup, network availability, basic framework for objective-C programming, location information and address book. Media layers deals with audio, video, 2D/3D graphics. Cocoa touch layer is responsible to develop graphical event driven applications in the iPhone OS shown in **Fig. 3.1**.

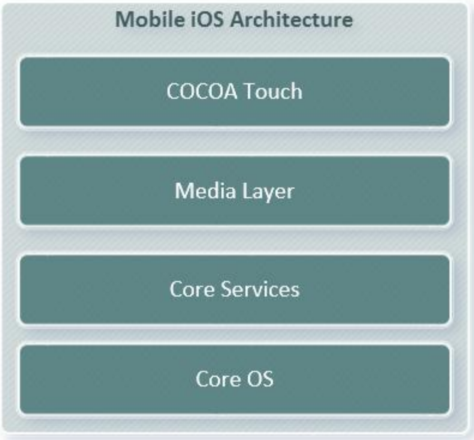

Fig. 3.1. Mobile iOS Architecture

3.2 Memory Management in iOS

Memory management in iOS was initially non-ARC (Automatic Reference Counting), where we must retain and release the objects. Now, it supports ARC and we don't have to retain and release the objects. XCode takes care of the job automatically in compile time [21] [22].

Memory management is a process of allocation of memory during the execution of program's or in runtime. Memory should be release/free when a task is completed [23]. As we know that a well written code utilizes minimum memory as possible. If we discuss memory management in objective-C language, it is observed that this is also a way of distributing ownership of limited memory resources between most of pieces of data and code.

There are some issues which are arising during the memory allocation process. IOS is freeing the memory or overwriting the memory that is still in use. Due to this, there are series of problems that occurs like memory corruption, application crashing, application corrupting the user data etc. Another very serious issue is, IOS is not releasing the data that is no longer in use which causes the memory leakage [24]. When allocated memory for some task is not freed up, even knowing that it will never be used again; called memory leak. Memory leaks increase the memory utilization against some process. As a result, poor performance of application occurs which results in application being terminated.

But, now iOS cover this problem in their operating system like iOS 7 so on. Now they are following new and updated architecture for the memory management which is shown in **Fig. 3.2**. New architecture is very efficient than previous one [25]. Because new architecture is releasing the memory after the completion of operation immediately or after some certain time, when an operation is taking too much time to execute.

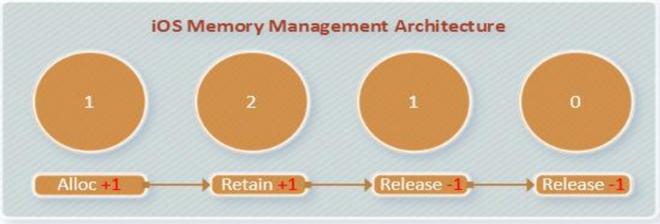

Fig. 3.2. iOS Memory Management Architecture

3.3 Security in iOS

iPhone offers different API's to overcome the security features for the application developers. iPhone OS used BSD and CDSA to implement the security features [26]. Low level features like file permissions is implemented using BSD Kernel which is a UNIX OS form. CSDA provides the higher-level functionality for encryption, security of data storage and authentication. SCDA is open source standard but it does not follow the standards of MAC programming conventions [27], that's why it is not directly accessible. When there is some security vulnerability found in service or application, the entire system may be compromised. This feature makes the difference between iOS and other operating systems like android and windows. In which, applications are shielded from the operating system and other services by running in the virtual machine [28].

3.4 Performance in iOS

iPhone is considered as the top smart phone in terms of performance. Because iOS has best architecture and memory management system [29]. Their applications are designed on specific and strict criteria due to security issues. iOS handles multiple operation at the same time without hanging. The best point of the iOS is, it uses minimum memory to run large application but in other operating system, they require more memory for performing larger operations.

4. Windows Operating System

Windows operating system developed by Microsoft to run on different mobile phones and tablets. Currently Windows mobile operating system is running on Nokia and Some of HTC mobiles. Windows mobile OS is based on the Win 32 API, and purpose of design is to provide similar services like provided in PC.

In section 4.1 we discuss the architecture of windows OS, while section 4.2, 4.3 and 4.4 briefly elaborates the techniques used in memory management, security concerns and performance analysis of windows OS respectively.

4.1 Windows OS Architecture

In windows mobile operating system, kernel is based on windows compact edition (CE) [30], an OS which is designed for handheld and embedded systems. In Windows CE, developer choose their own required functionality. Size of the OS including kernel is few hundreds KB, but developers can add their own choice components like web browser or to support .NET framework. The major difference between Windows CE and Windows Mobile is that there are some group of components that are fixed and used by Microsoft so that APIs are reliable between all Windows Mobile Phone devices. But remember, APIs are always not same, it is required by all mobile phones to manage minimum set of functionalities, but the mobile phone manufactures include the additional APIs in the operating system image freely.

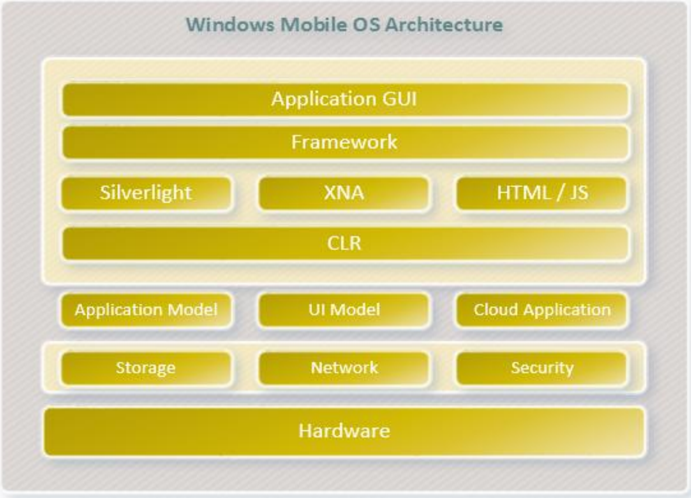

Fig. 4.1. Windows Mobile OS architecture

In **Fig. 4.1**, Windows mobile architecture is divided into four layers. First layer is used to visualization. Second layer is used for modeling and handling cloud application. Third layer is used for storage, network and security etc. Fourth layer is consisting on hardware.

4.2 Memory Management in Windows

Windows is one of the top leading company among the technological devices. Windows phones are not much popular as their other device like Windows servers, windows 10 OS for laptops and PC's etc. As one of the top ranked company in technology, Microsoft is spending too much in their mobile industry. There are many issues present in the Windows mobile OS. The main issue in windows mobile is stop the execution of an application in the middle of its working as well as start the application from scratch when turnoff the screen for more than 5 minutes. Other issues include crashes the application, hanging the application during execution etc. Windows memory plan is yet simple but overall it can be a flexible plan if they will overcome these issues. As stated above there are some of requirement for the good memory application. As a solution, Microsoft introduces the paging, also known as virtual memory [31]. In this concept, if main memory is filled with specific data of application then it takes someplace from the permanent memory. So many of the new applications are designed on the basis on virtual memory concepts. But still some of the applications which are still not compatible with memory technology, and they are creating issues for the memory of windows phones. Usually windows memory management cycle is perfect in looking if they will overcome the issues mentioned above. In new Windows phone OS, if a user is working on some application, Windows freeze the background application and give the full support to the front running app. This technique may save the memory space and speed up the current application in the memory for some time.

4.3 Security in Windows

As mentioned above, Windows Mobile operating system is very similar to desktop counter in different ways. Just like android, Windows Mobile phone developer have full freedom to develop the application on low level using the different low-level APIs. In terms of security measures, the developed applications are safe and secured and not interrupting the system process. The application is not allowed to access the process space of another process directly [32]. But if there is need of this, it can be accessible via DLL injections mechanism. By default, process space is designed as secured from errors and applications too. But it is not enough to protect applications from malicious applications. There will be restriction if there is new process initialized as trusted process. However, it is possible that the applications which access the trusted APIs required to be signed before access is allowed, but this technique depends on the implemented security plan.

4.4 Performance in Windows

Windows phones spent too much cost on increasing the performance. In earlier versions of Windows phones operating systems, some of the applications running in the memory automatically closed [33] [34]. But with new windows phone operating system, this issue is resolved to some extent but still they are not too much faster with compare to Android and iOS. There interface is not organized and user friendly. System settings are difficult to find.

5. Comparison: Android and Windows

While both android and windows manage the memory effectively according to the need, but there are some noteworthy differences. To check the android memory management, an app starts multiple foreground processes to allocate memory resources while keeping the default limit of heap allocation to all the applications in order to prevent using all the available memory [35]. Android using its “low memory killer” can kill any process or threads associated to it will also be destroyed at any time to release the resources as needed. Android system also tries to maintain application process for as long as possible which results in saving the battery, but it needs to close old applications in order to give priority of resources to the new applications [36]. On the other hand, windows support virtual memory and can hold more processes than android which is using only RAM, making it more efficient in terms of large applications handling. Both the OS shows the lag in application handling as the demand of memory increases. Windows uses virtual memory when the RAM is full, making the performance degradation, as programs on virtual memory will run slowly, but android kills the process on RAM which are idle or not being used for long period to free up the memory for new processes. In short, both OS reacts differently, according to the need. No one is running high performance computing. Following table show some comparisons of different parameters in android and windows OS.

6. Comparison: Android and iOS

Both the OS handles the memory efficiently according to the resources, but iOS uses less amount of memory than android to compete against all the resources for different applications [37]. There are several reasons behind this scenario. In general, when the user is not active, the apps running move to the background state. Background state further leads to the suspension of an application, which will result in the improvement of battery life and also free up the resources of the system to be used for new fore-ground applications. Most of the applications continuously run in the background, like the music app which need to continue audio playing on lock screen, downloading of any process might also continue in the background state of the system. In terms of background running of applications, iOS does it efficiently and without draining system resources or the battery life [38] [39]. Following are the reasons behind it:

- Applications that take short time to finish in the fore-ground can take that time to finish first and then move to the background state
- Applications that start downloading in the fore-ground handles all the management to the system, which will allow the system to continue the downloading while the apps will suspend or terminate
- Applications that need to run in the background (like music app) will declare its support to one or more background modes

Both the systems try to avoid doing background processing unless doing so will improve the performance, which iOS handles efficiently than android. When an application is not being used or the user locked the device, that app is not meaningful to run in background, again iOS understands it correctly but android does not.

Android use the method for garbage collection (GC) which is a slow process of releasing memory while iOS did not [40], even though GC is entirely pointless when ARC is being used by iOS. Every object in iOS has a count called the “retain count”, during the lifecycle this count will increase and decrease, but once the count becomes zero, the object will be destroyed immediately making it unnecessary for any garbage collector.

There is also the reason of handling the UI rendering [41], which iOS handles perfectly but android does not, which creates bit more lag over iOS. While an iOS UI rendering happens in separate thread with real time priority, on android this happens in main thread with normal priority. When other apps will take over the processor resources, this will surely hurt the UI interactions. Following table shows comparison of some parameters of iOS and Android OS.

7. Comparison: Windows and iOS

In the comparisons of windows and IOS, there is big difference between both operating systems. iOS only supports their own company mobile phones which is iPhone. But Windows phone operating system is supported by the Nokia Lumia series and some of HTC handsets. iOS is increasing their performance using the patches in case of any issue occurs. But Windows phone operating system developer take some time to deliver updates. iOS is handling their background processes very efficiently with the freezing of background process and just run the current process used by the user [42]. But in windows operating system, some of the process automatically terminated after some time or some of the application loaded from scratch after

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3344v1 | CC BY 4.0 Open Access | rec: 13 Oct 2017, publ: 13 Oct 2017

some time, which is the biggest issue in windows phone operating system. iOS has better interfaces w.r.t Windows phones operating systems. iOS mostly uses their own utilities. But windows OS take some utilities from the third parties. Windows phone OS uses mobile phone internal memory for the increasing processing speed. But iOS didn't use the internal memory for the processing. iOS have a great quality which is, they utilized less memory but perform more faster processing with compare to Windows OS.

8. Comparison: Windows, iOS and Android

Now have a look at the previous versions of IOS, Windows and Android Memory Management schemes. In 2014, Apple launched its iPhone 6 and iPhone 6 plus. Both smartphones launched by the apple having 1 GB of ram. But in compare to other mobile phones, Samsung launched its S5 handset with 2 GB ram, LG launched G3 with 2 and 3 GB RAM respectively, Google launched Nexus 6 with 3 GB ram and Nokia launched Lumia series with Windows 7 OS having 512 MB of ram. In 2015, Apple launched 6S and 6S Plus with 2GB ram, Samsung released S6 and S6 edge with 3GB and S6 Edge+ with 4GB ram, LG launched G4 with 3GB and V10 with 4GB ram, Google launched Nexus 5X with 2 GB ram and Nexus 6P with 3 Gb ram and Nokia released Lumia series with Windows 8 OS having 1GB ram. In 2016, Apple launched iPhone 7 and iPhone 7 Plus with 2GB and 3Gb ram respectively, Samsung launched S7 and S7 Edge with 4Gb ram, LG launched G5 and V20 with 4 GB ram, Google launched Pixel and Pixel XL with 4GB ram and Nokia released in Lumia with Windows 8.1 and 10 OS having 2 and 3 GB of ram.

If we analyze the memory of each mentioned handset, then there is clear difference between iOS handsets with Android and Windows mobile phones. iPhone uses iOS only and almost all other smartphones uses the android platform except the Windows phone [43] [44]. iOS is most successful in memory management because it uses less memory at the same time while utilizing heavy use of it, giving excellent response to the users running heavy applications smoothly and quickly [45]. On the other hand, windows mobiles are also using less memory but they are not satisfying their users resulting in application crashes and hand issues. Also, iOS works efficiently when it comes to background processes and their handling [46]. It freezes the background processes in order to give all the resources to the fore ground processes, making it fast and quick responsive while this capability lack in both android and windows OS. In compare to android handsets, they are using more memory w.r.t to iOS and windows but the performance is somehow relatively same in terms of user satisfaction [47].

When it comes to security, iOS uses its own utilities and has its own store for its applications and stuff, which makes it safe from different threats and viruses, but in case of android and windows, security lacks as they use third party applications for utilities [48]. Android is open source platform which makes it more dangerous, as many viruses and worms can attack it easily but windows and iOS are not open source.

The interface of both android and iOS is user friendly making it an open option for any type of user to use it freely [49], while windows OS on mobile phones is not so user friendly and this shows a performance degradation for the users and applications. iOS understands quickly about which application need more resources and processor [50], resulting in efficient use of resources for each application, but these features does not exist in both android and windows OS of mobile phones. So, keeping in view of all these arguments, iOS is much better OS when we talk about memory management, security and performance in mobile phones as compare to android and windows OS in **Table 1**.

Table 1. Comparison of android, iOS and windows OS

Parameters	**(Memory Management, Security and Performance) Comparison**		
	Android	**iOS**	**Windows**
Memory Management			
Memory usage	High	Low	High
Memory used for App handling	RAM	RAM	RAM + VM
Process running in background	Not Efficiently	Efficiently	Not Efficiently
Use of Garbage Collector	Yes	No	Yes
Background Processes	Do not freeze	Freeze	Suspend
To increase process speed	Uses internal memory	Don't use internal memory	Uses internal or virtual memory
Interface	User Friendly	User Friendly	Not User Friendly
Increase in Memory demand	Lag in app handling	No lag in app handling	Lag in app handling
Shortage of Memory	May kill some processes	Freeze background processes	Uses Virtual Memory
Capable of loading large number of apps	No	No	Yes
Security			
Arrival of new process	May kill existing process	Freeze some processes	No other processes will be affected
Utilities used	Own and third party	Own	Third Party Mostly
Issue Occurrence	Use patches	Use patches	Deliver updates
Rooting	Allowed	Not allowed	Not allowed

Table 2 shows recommendation for all the mobile users against different features. It states that which operating system is best in which scenario according to different features w.r.t memory management of applications, security and performance. **Table 2** also shows the ranking of each operating system according to the feature so that, the best operating system has 'First' ranking, the second best has 'Second' ranking and the third best has 'Third' ranking respectively.

Table 2. Ranking of OS's Against Different Features

Features	**Android**	**Windows**	**iOS**
Affordability, Interface and apps	First	Third	Second
Applications speed	Second	Third	First
App store usability	Second	Third	First
Alternative app stores and side-loading	First	Third	Second
Battery life and management	First	Third	Second
OS updates	Second	Third	First

Customizability	First	Second	Third
Calls and Messaging	Second	Third	First
Email	First	Third	Second
Voice Assistants	Second	First	Third
Connectivity	First	Second	Third
Sharing data	First	Second	Third

9. Conclusion

After studying the techniques of memory management and security concerns, performance of popular operating systems like android, windows and iOS is evaluated. Our main concern was to compare the different operating systems memory utilizations, which operating system use the memory efficiently in terms of processes loaded in it, application launch time and switching between two apps. Security concerns of different operating systems as well as their performance in terms of running the processes is also compared. Performance of each OS is analyzed keeping in mind memory management and security issues. After thorough review, we can clearly say that iOS was better when compare to windows and android devices in terms of total memory, effective utilization of that memory, operating system security and in overall performance.

References

[1] Ahmad, M. S., Musa, N. E., Nadarajah, R., Hassan, R., & Othman, N. E. (2013, July). Comparison between android and iOS Operating System in terms of security. In *Information Technology in Asia (CITA), 2013 8th International Conference on* (pp. 1-4). IEEE.

[2] Armel, A. S. R., & Thavavel, V. (2013, December). Ghost encryption: Mobile data security model encrypting data before moving it to the cloud service provider. In *Advanced Computing (ICoAC), 2013 Fifth International Conference on* (pp. 512-516). IEEE.

[3] Aron, L., & Hanacek, P. (2015, March). Overview of security on mobile devices. In *Web Applications and Networking (WSWAN), 2015 2nd World Symposium on* (pp. 1-11). IEEE.

[4] Kocoloski, B., & Lange, J. (2016). Lightweight Memory Management for High Performance Applications in Consolidated Environments. *IEEE Transactions on Parallel and Distributed Systems*, *27*(2), 468-480.

[5] Long, L., Liu, D., Liang, L., Zhu, X., Zhong, K., Shao, Z., & Sha, E. H. M. (2016). Morphable Resistive Memory Optimization for Mobile Virtualization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *35*(6), 891-904.

[6] Diener, M., Cruz, E. H., Alves, M. A., Navaux, P. O., Busse, A., & Heiss, H. U. (2016). Kernel-based thread and data mapping for improved memory affinity. *IEEE Transactions on Parallel and Distributed Systems*, *27*(9), 2653-2666.

[7] Doriguzzi-Corin, R., Siracusa, D., Salvador, E., & Schwabe, A. (2016, April). Empowering network operating systems with memory management techniques. In *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP* (pp. 740-744). IEEE.

[8] Liu, L., Li, Y., Ding, C., Yang, H., & Wu, C. (2016). Rethinking Memory Management in Modern Operating System: Horizontal, Vertical or Random? *IEEE Transactions on Computers*, *65*(6), 1921-1935.

[9] Kocoloski, B., & Lange, J. (2014, May). HPMMAP: Lightweight memory management for commodity operating systems. In *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International* (pp. 649-658). IEEE.

[10] Kim, H., Lim, H., Manatunga, D., Kim, H., & Park, G. H. (2015). Accelerating Application Start-up with Nonvolatile Memory in Android Systems. *IEEE Micro*, *35*(1), 15-25.

[11] Nomura, S., Nakamura, Y., Sakamoto, H., Hamanaka, S., & Yamaguchi, S. (2014, October). Improving choice of processes to terminate in Android OS. In *Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on* (pp. 624-625). IEEE.

[12] Vimal, K., & Trivedi, A. (2015, December). A memory management scheme for enhancing performance of applications on Android. In *Intelligent Computational Systems (RAICS), 2015 IEEE Recent Advances in* (pp. 162-166). IEEE.

[13] Singh, S., Nivangune, A., Kumar, S., Kumar, R., Joshi, P., & Patel, D. (2016, May). Extending app installable memory in Android smartphones. In *Proceedings of the International Workshop on Mobile Software Engineering and Systems* (pp. 237-240). ACM.

[14] Santhanakrishnan, G., Cargile, C., & Olmsted, A. (2016, October). Memory leak detection in android applications based on code patterns. In *Information Society (i-Society), 2016 International Conference on* (pp. 133-134). IEEE.

[15] Sylve, J., Case, A., Marziale, L., & Richard, G. G. (2012). Acquisition and analysis of volatile memory from android devices. *Digital Investigation*, *8*(3), 175-184.

[16] *Gronli, T. M., Hansen, J., Ghinea, G., & Younas, M. (2014, May*). Mobile application platform heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS. In *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on (pp. 635-641). IEEE.*

[17] Hernandez, I. M. T., Viveros, A. M., & Rubio, E. H. (2013, March). Analysis for the design of open applications on mobile devices. In *Electronics, Communications and Computing (CONIELECOMP), 2013 International Conference on (pp. 126-131). IEEE.*

[18] Schuler, A., & Franz, B. (2013, April). Rule-based generation of mobile user interfaces. In *Information Technology: New Generations (ITNG), 2013 Tenth International Conference on (pp. 267-272). IEEE.*

[19] Nguyen, T. A., Rumee, S. T., Csallner, C., & Tillmann, N. (2012). An experiment in developing small mobile phone applications comparing on-phone to off-phone development *(pp. 9-12). IEEE.*

[20] Zhao, T., Zhang, G., & Zhang, L. (2014, December). An overview of mobile devices security issues and countermeasures. In *Wireless Communication and Sensor Network (WCSN), 2014 International Conference on* (pp. 439-443). IEEE.

[21] Adibi, S. (2014, May). Comparative mobile platforms security solutions. In *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on* (pp. 1-6). IEEE.

[22] *Holder, E., Shah, E., Davoodi, M., & Tilevich, E. (2013, November). Cloud twin: Native execution of Android applications on the Windows Phone. In Automated Software Engineering (ASE), 2013 IEEE/ACM 28th International Conference on (pp. 598-603). IEEE.*

[23] Wijaya, R., Wuryandari, A. I., & Nugraha, H. C. (2012, September). OpenSURF performance in windows phone 7. In *System Engineering and Technology (ICSET), 2012 International Conference on (pp. 1-5). IEEE.*

[24] Willocx, M., Vossaert, J., & Naessens, V. (2016, May). Comparing performance parameters of mobile app development strategies. In *Proceedings of the International Workshop on Mobile Software Engineering and Systems (pp. 38-47). ACM.*

[25] Ahn, H., Cho, S., Na, H., & Han, H. (2009). Access pattern based stream buffer management scheme for portable media players. *IEEE Transactions on Consumer Electronics*, *55*(3).

[26] Kang, H., Park, J., & Bahn, H. (2010). LBM: a low-power buffer management policy for heterogeneous storage in mobile consumer devices. *IEEE Transactions on Consumer Electronics*, *56*(4).

[27] Lim, G., Min, C., & Eom, Y. (2013). Virtual memory partitioning for enhancing application performance in mobile platforms. *IEEE Transactions on Consumer Electronics*, *59*(4), 786-794.

[28] Kim, S. H., Jeong, J., & Lee, J. (2014). Selective memory deduplication for cost efficiency in mobile smart devices. *IEEE Transactions on Consumer Electronics*, *60*(2), 276-284.

[29] Kwon, Y., Lee, S., Yi, H., Kwon, D., Yang, S., Chun, B. G., ... & Paek, Y. (2015). Mantis: Efficient predictions of execution time, energy usage, memory usage and network usage on smart mobile devices. *IEEE Transactions on Mobile Computing*, *14*(10), 2059-2072.

[30] Ryu, S., Lee, K., & Han, H. (2015). In-memory write-ahead logging for mobile smart devices with NVRAM. *IEEE Transactions on Consumer Electronics*, *61*(1), 39-46.

[31] Guo, W., Chen, K., Feng, H., Wu, Y., Zhang, R., & Zheng, W. (2016). MARS: Mobile Application Relaunching Speed-Up through Flash-Aware Page Swapping. *IEEE Transactions on Computers*, *65*(3), 916-928.

[32] Sun, H., Sun, K., Wang, Y., & Jing, J. (2015). Reliable and Trustworthy Memory Acquisition on Smartphones. *IEEE Transactions on Information Forensics and Security*, *10*(12), 2547-2561.

[33] Jeong, J., Kim, H., & Lee, J. (2016). Transparently Exploiting Device-Reserved Memory for Application Performance in Mobile Systems. *IEEE Transactions on Mobile Computing*, *15*(11), 2878-2891.

[34] Zhong, Kan, et al. "Energy-efficient in-memory paging for smartphones." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35.10 (2016): 1577-1590.

[35] Lin, Y. J., Yang, C. L., Li, H. P., & Wang, C. Y. M. (2016). A Hybrid DRAM/PCM Buffer Cache Architecture for Smartphones with QoS Consideration. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, *22*(2), 27.

[36] Yoon, S. K., Youn, Y. S., Park, K., & Kim, S. D. (2017). Mobile Unified Memory-Storage Structure Based on Hybrid Non-Volatile Memories. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, *13*(3), 40.

[37] Kim, S. H., Jeong, J., Kim, J. S., & Maeng, S. (2016). SmartLMK: a memory reclamation scheme for improving user-perceived app launch time. *ACM Transactions on Embedded Computing Systems (TECS)*, *15*(3), 47.

[38] Huguenard, B. R., Lerch, F. J., Junker, B. W., Patz, R. J., & Kass, R. E. (1997). Working-memory failure in phone-based interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *4*(2), 67-102.

[39] Enck, W., Ongtang, M., & McDaniel, P. (2009). Understanding android security. *IEEE security & privacy*, *7*(1), 50-57.

[40] Li, Q., & Clark, G. (2013). Mobile security: a look ahead. *IEEE Security & Privacy*, *11*(1), 78-81.

[41] Jeong, J., Kim, H., Hwang, J., Lee, J., & Maeng, S. (2013). Rigorous rental memory management for embedded systems. *ACM Transactions on Embedded Computing Systems (TECS)*, *12*(1s), 43.

[42] Hu, W. C., Zuo, Y., Kaabouch, N., & Yang, H. J. (2009, May). Mobile Data Protection Using Handheld Usage Context Matching. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on* (pp. 594-599). IEEE.

[43] Joshi, J., & Parekh, C. (2016, April). Android smartphone vulnerabilities: A survey. In Advances in Computing, Communication, & Automation (ICACCA)(Spring), International Conference on (pp. 1-5). IEEE.

[44] Tilson, D., Sorensen, C., & Lyytinen, K. (2012, January). Change and control paradoxes in mobile infrastructure innovation: The Android and iOS mobile operating systems cases. In System Science (HICSS), 2012 45th Hawaii International Conference on (pp. 1324-1333). IEEE.

[45] Delac, G., Silic, M., & Krolo, J. (2011, May). Emerging security threats for mobile platforms. In MIPRO, 2011 Proceedings of the 34th International Convention (pp. 1468-1473). IEEE.

[46] Mayer, D. A. (2016, May). idb: a tool for blackbox iOS security assessments. In Proceedings of the International Workshop on Mobile Software Engineering and Systems (pp. 281-282). ACM.

[47] Tracy, K. W. (2012). Mobile application development experiences on Apple's iOS and Android OS. Ieee Potentials, 31(4), 30-34.

[48] Masruroh, S. U., & Saputra, I. (2016, April). Performance evaluation of instant messenger in Android operating system and iPhone operating system. In Cyber and IT Service Management, International Conference on (pp. 1-6). IEEE.

[49] Lee, J. E., Park, S. H., & Yoon, H. (2015, April). Security policy based device management for supporting various mobile os. In Computing Technology and Information Management (ICCTIM), 2015 Second International Conference on (pp. 156-161). IEEE.

[50] Jing, Y., Ahn, G. J., Zhao, Z., & Hu, H. (2015). Towards automated risk assessment and mitigation of mobile applications. IEEE Transactions on Dependable and Secure Computing, 12(5), 571-584.