International Journal of Computer & Mathematical Sciences
IJCMS
ISSN 2347 – 8527
Volume 5, Issue 7
July 2016

A Comparative Study of Mobile Operating Systems with Special Emphasis on Android OS

Aaditya Jain[1]
[1]M.Tech Scholar, Department of Computer Science & Engg.,
R. N. Modi Engineering College, Rajasthan Technical University, Kota, Rajasthan, India

Samridha Raj[2]
[2]Department of Computer Science & Engg.,
R. N. Modi Engineering College, Rajasthan Technical University, Kota, Rajasthan, India

Dr. Bala Buksh[3]
[3]Professor, Department of Computer Science & Engg.,
R. N. Modi Engineering College, Rajasthan Technical University, Kota, Rajasthan, India

ABSTRACT

In today's world, everybody from a lay man to an industrialist is using a mobile phone. Therefore, it becomes a challenging factor for the mobile industries to provide best features and easy to use interface to its customer. Due to rapid advancement of the technology, the mobile industry is also continuously growing. There are many mobile phones operating systems available in the market but mobile phones with android OS have now become domestic product which was once extravagant product. The reason towards this change is attributed to its varied functionality, ease of use and utility. Increased usage of smart phone has led towards higher concerns about security of user- private data. Due to android as an open source mobile platform, user can easily install third party applications from markets and even from unreliable sources. Thus, Android devices are a soft target for privacy intrusion. Whenever the user wants to install any application, firstly it's the description and the application screenshots which provides an insight into its utility. The user reviews the description as well as a list of permission requests before its installation. As the types and rate of malicious attacks increases, the difficulty of examining in advance whether an app is malicious or not through its descriptions has increased manifolds. In this paper we have reviewed and examined android software stack and compared smart phone based operating system like Android, iOS, Symbian, Windows phone, Blackberry.

Keywords— Mobile OSs, Android, iOS, Symbian, Windows, Blackberry.

I. INTRODUCTION

A Smartphones are now participating nearly in each and every sphere of life like business, education, workplace and healthcare. The Worldwide Mobile Communications Device Open Operating System Sales (WMCDOOS) provides total market of 104,898 to End Users by OS [3]. There are over 1.3 million active applications [4] in Google Play App Store. Android is the first open source, Linux-based and modern mobile handset platform. Google developed it for handset manufacturers like T-Mobile, Sprint Nextel, Google, Intel, Samsung, etc. [1]. It offers to consumers a richer, less expensive, better mobile experience and various features like 3D, SQLite, Connectivity, WebKit, Dalvik and FreeType etc. Since android provides open source operating system; users and developers can get source code but only under the rules and conditions. Whenever the user wants to install any application, firstly its description as well as a list of permission requests is provided with an opportunity for review before its installation or cancel the installation if he or she finds that the permissions are too many or objectionable. The android operating system has its own well established android permission model, but intruders can supplements them by allowing the components to be changed within and across the applications through Intent communication mechanism due to which it has susceptibility for attacks by malwares [5]. Android open source platform requires strong and complex security architecture to ensure security of user private data, personal information, application and network, but it has few constraints for developers which raises the security risk for the end users [2].

In this paper section 2 presents android architecture with brief description of its software stack, section 3 present the study of another popular mobile operating systems with their architectures, section 4 compares the latest smart phone operating system like android, iOS, blackberry, Symbian, windows phone. The comparison

International Journal of Computer & Mathematical Sciences
IJCMS
ISSN 2347 – 8527
Volume 5, Issue 7
July 2016

between different smart phone operating system is done using different parameters; section 5 includes conclusions and future scope of the research work.

II. Software Stack of Android Mobile Operating System

Android OS is architected in the form of different layers of stacked as software that comprises android applications, an operating system, android run-time, middleware, services and libraries. Each layer of the stack, and the corresponding elements within each layer, are tightly integrated and provides different kind of services to the layer just above it as well as the optimal application development and execution environment for mobile devices.

The Software stack of android consists of different layers that provide different services to layer just above it are shown in figure 1.

- Linux Kernel- heart of whole system
- Libraries and Android Runtime
- Application Framework
- Android Applications.

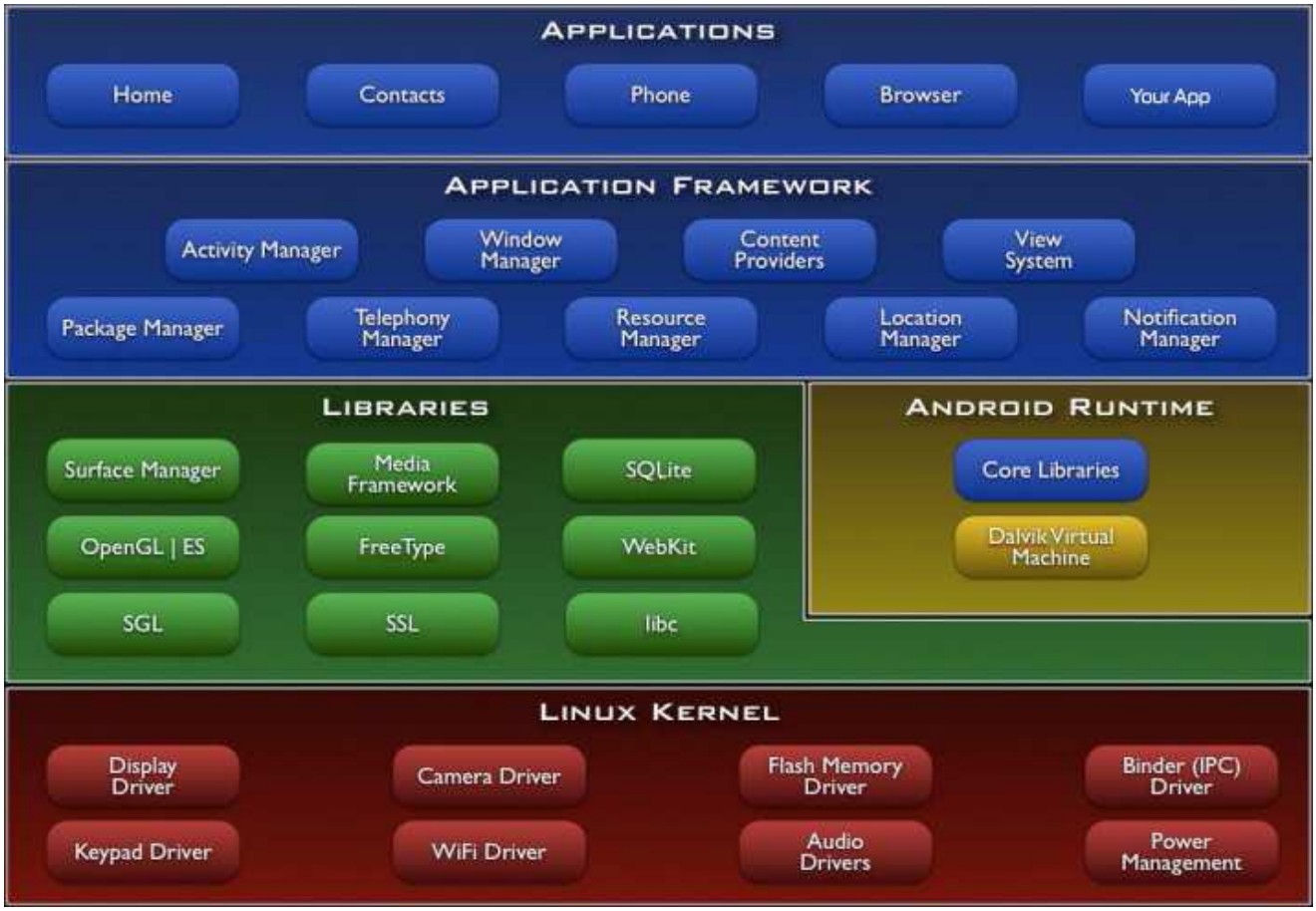

Fig. 1 Software stack of android [1], [6]

2.1 Linux Kernel: Heart of the Whole System

At bottom of the Software stack of Android, there is a Linux kernel. It acts as the heart of the whole system. It provides various functionalities like memory management, process management, device management, security settings etc. in android system and all the essential device drivers for the hardware with which it interacts [1].

2.2 Native Libraries

On the top of the kernel layer there is a set of libraries including surface manager that composes windows on the screen, Open GL|ES for 3D Library, SGL for 2D Graphics, Media Framework to play and recording of various audio, video and picture formats, Free Type for Font Rendering, WebKit Browser Engine, well known libc for System C libraries, SQLite relational database for storage, Open SSL internet security library etc. These native libraries are based upon C or C++ language.

2.3 Android Runtime

Located on the same level as the native libraries, the Android runtime is the third section of the architecture and available on the second layer from the bottom. It includes a set of core Java libraries that enables Android application developers to write Android applications using standard Java programming language. It also includes ART (Android Runtime) [7]. It is a similar to DVM (Dalvik Virtual Machine) specially designed and optimized for Android. Each process is executed in a virtual machine separately. It operates on DEX files and running dex byte codes. It provides Ahead-of-Time (AOT) compilation, improved garbage collection, improved debugging and development, security, isolation, memory management, fast performance and threading support. It helps user to execute multiple applications at the same time.

2.4 Application Framework

On the top of Native libraries and android runtime layer, there is application framework layer. It provides many packages of higher-level services to application that collectively form the environment within which they are constructed from reusable, interchangeable and replaceable components. It provides the functions of phone like location management, data sharing, resource management etc.

The packages present are as given below:

Activity manager: It controls and manages the activity lifecycle of applications.

Resource manager: It manages and provides access to non-code embedded resources such as graphics, strings, color settings and user interface layouts.

Notification manager: It allows all applications to show custom alerts in status bar and notifications to the user.

Location manager: When user enters or leaves a particular geographical location, it triggers alerts about location changes using GPS or cell tower.

Package manager: The system by which applications are able to retrieve the data about other applications currently installed on the device.

Telephony manager: It manages and enables to access voice calls, network connection settings, status and subscriber information service in our application.

Window manager: An extensible set of creative views and layouts is used to create application user interfaces.

Content Provider: It is the system by which it enables and manages data sharing between applications [1].

2.5 Android Applications

The android applications are at the topmost layer of the Android software stack. These comprise both the native applications and the third party applications. The native applications provide the basic Android implementation such as SMS client app, Dialer, Web browser and Contact manager. The third party applications are further installed by the developers, programmers while debugging/testing and user after purchasing the device.

2.6 Android Versions

The first release of android was in September 23, 2008 and has numerous updates which incrementally improves the operating system by adding new features and fixing bugs. Table 1 shows the different versions of Android with their code name, release date and API level [3].

International Journal of Computer & Mathematical Sciences
IJCMS
ISSN 2347 – 8527
Volume 5, Issue 7
July 2016

Table 1: Different versions of android

Version	Code Name	Release Date	API Level
6.1	Marshmellow	May 15, 2015	21
5.1	Lollipop	October 15, 2014	20
4.4	KitKat	October 31, 2013	19
4.3.x	Jelly Bean	July 24, 2013	18
4.2.x		November 13, 2012	17
4.1.x		July 09, 2012	16
4.0.3-4.0.4	Ice Cream Sandwich	December 16, 2011	15
3.2	Honeycomb	July 15, 2011	13
2.3.3-2.3.7	Gingerbread	February 09, 2011	10
2.2	Froyo	May 20, 2010	8

III. Other Popular Mobile Operating Systems

3.1 iOS

Iphone 6 features an A8 chip built on second generation 64-bit desktop-class architecture. It is enhanced by an M8 motion coprocessor that measures activity from advanced sensors. Earlier versions were developed from ARM processors. In iPhone, Hardware refers to the physical chips soldered to the iPhone's circuitry. The actual processor falls under this layer, but the instruction set and in-memory descriptor tables are contained within the "processor" layer [8]. Firmware refers to the chip-specific code that is either contained with memory in/around the peripheral itself, or within the drive for said peripherals. iPhone OS is the kernel, drivers, and services that comprise of the iPhone Operating System. It sits between the user space and hardware. Objective-C runtime is comprised of both the Objective-C dynamically-linked runtime libraries, as well as the underlying C libraries. Frameworks/API layer has API calls which are Apple-distributed headers with the iPhone SDK, with some dynamic linking occurring at runtime. These reside on top of the Objective-C runtime, as many of these are written in Objective-C [9]. The application stored in iPhone has to be purchased through the application store [10]. This application was compiled to native code by the Apple-distributed iPhone compiler, and linked with the Objective-C runtime and C library by the linker. The application also runs entirely within the user space environment set up by the iPhone OS.

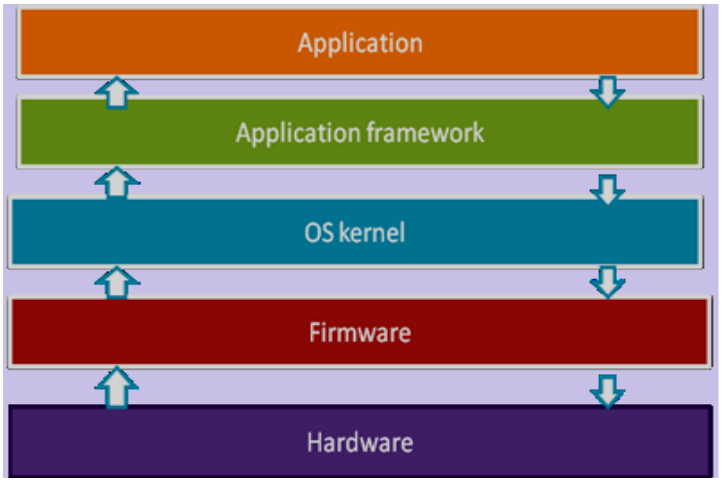

Fig. 2 Architecture of iOS mobile OS

International Journal of Computer & Mathematical Sciences
IJCMS
ISSN 2347 – 8527
Volume 5, Issue 7
July 2016

3.2 Symbian

Symbian mobile operating system with libraries, UI frames works and common tools. It is descendant of Psions EPOC and run exclusively on ARM processors. Symbian OS was built to follow three design rules in order to support extended always on operation [11].

- The integration and security of user data is important.
- User time must not be wasted.

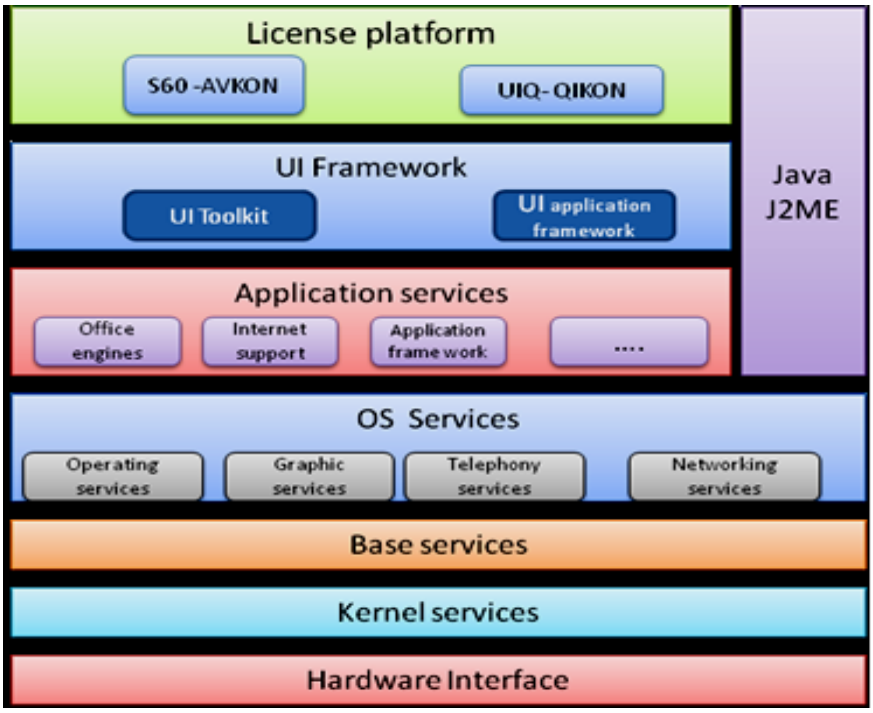

Fig 3 Architecture of symbian mobile OS

For hardware, the OS is optimized for low power battery based and for ROM based systems. The later version of Symbian has a real time Kernel and a platform security model (version 8 and 9). There is a strong emphasis on conserving resources, using Symbian specific programming features such as descriptors and clean up stack. All Symbian OS programming is event-based, and the CPU is switched off when applications are not directly dealing with an event. Similarly, the OS approach to thread vs. process is driven by reducing over heads. It supports fast real time response, that a single core executes both the user application and signaling stack in a mobile phone. This feature is not available in Linux also. This feature makes Symbian OS phones to become smaller cheaper and more power efficient.

The Symbian OS model contains the following layers [13].

- UI frame work layer.
- Application services layer containing the following services:
  - Generic OS services.
  - Communications services.
  - Multimedia and graphic services.
  - Connectivity services.
- Base service layer.
- Kernel services and hardware interface layer.

The base services layer is the lowest level, reachable by user side operation. It includes the file server and user library, the plug in framework which manages all plugs in store, control, Data Base Management System (DBMS) and cryptographic services. It also includes the Text window server and the Text shell, complete

International Journal of Computer & Mathematical Sciences
IJCMS
ISSN 2347 – 8527
Volume 5, Issue 7
July 2016

functional port can be created from this, without the need for any higher level services. Symbian OS has a Microkernel architecture that provides robustness availability and responsiveness. It contains a scheduler, memory management and device drivers. Other services like networking telephony and file system support are present in the OS services layer or base services layer. The inclusion of device drivers means the kernel is not a true microkernel but a nanokernel containing only the basic primitives and supporting an extended kernel to implicate any other abstractions Symbian OS is designed for compatibility with other devices. There is a large networking and communication subsystem for EPOC telephony. The subsystem also contains code for short range communication like Bluetooth, IrDA (Infra Red Data Association) and USB. In Symbian OS the actual UI are maintained by third parties. A JVM (Java ME) is also included above the OS services layer.

3.3 Windows

In Windows CE.NET based architecture, the ROM stores the entire OS as well as the applications that come with the system, like Pocket Word, Pocket Excel. If a module is uncompressed ROM-based modules are executed in place. If the ROM based module is compressed, then first decompressed and then paged into RAM. All read and write data are loaded into RAM1 [9]. The option to enable compression in ROM is controlled by the original equipment manufacturer (OEM) executing programs directly from ROM. The RAM on a Windows CE device is dividing into two areas one is object store and the program memory [12]. The object store resembles a permanent virtual RAM disk. Data in object store is retained when the system is suspended or soft reset, and devices typically have a backup power supply for the RAM to preserve data if the main supply is interrupted temporarily. As a preemptive, multitasking OS, CE supports up to 32 processes running simultaneously within the system. The actual number of additional threads is limited only by the available system resources.

The windows CE.NET scheduler maintains a priority list of each process and thread in the OS. Each process can contain multiple threads, and each of these threads composes a path of execution. The scheduler controls the order in which these different paths of execution are sequenced and allows them to interact in a predictable fashion. Scheduler performs its work both from the kernel and in a predefined scheduling mechanism.

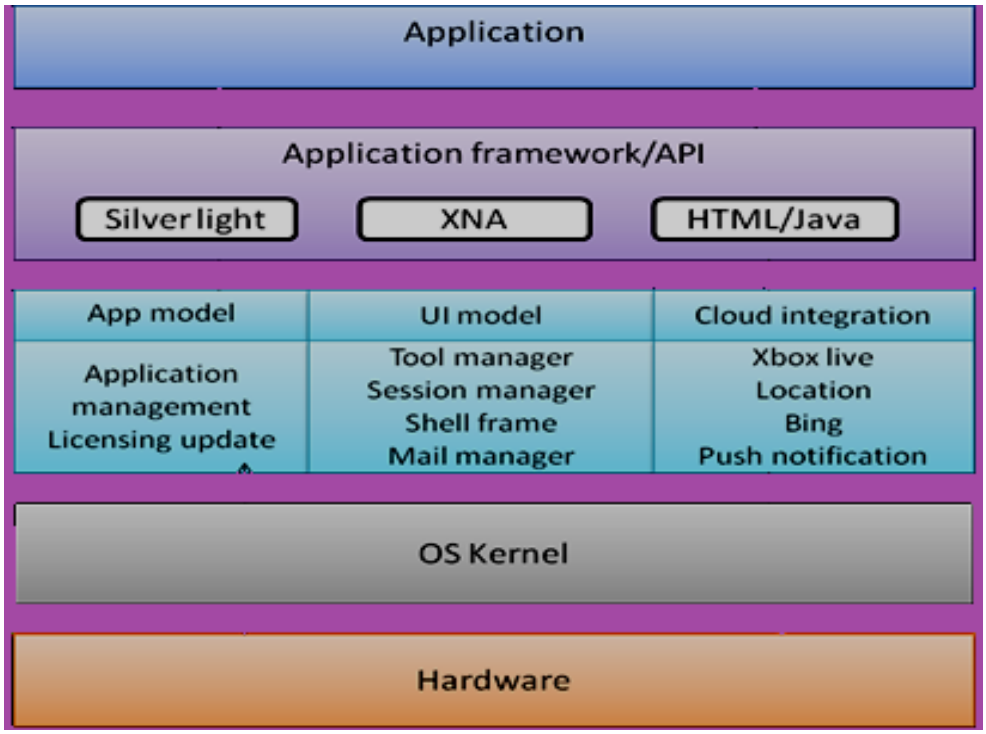

Fig. 4 Architecture of windows mobile OS

International Journal of Computer & Mathematical Sciences
IJCMS
ISSN 2347 – 8527
Volume 5, Issue 7
July 2016

3.4 Blackberry

Within Blackberry devices Java is integrated tightly and represents the only possible programming language for the Blackberry device. So it is possible to write native code for Blackberry device [11]. Blackberry devices have a proprietary Java virtual machine (JVM) which offers both Java ME standard features as well as Blackberry specific Java API extensions. Java is used for third party application development including email, contacts, calendar, web browser, etc. It also supports a large set of additional Java APIs that are not part of the standard Java ME specification for greater support of any particular hardware. Of course it is always not necessary to use these specific classes as greater features and functionality often provided within standard Java ME specification.

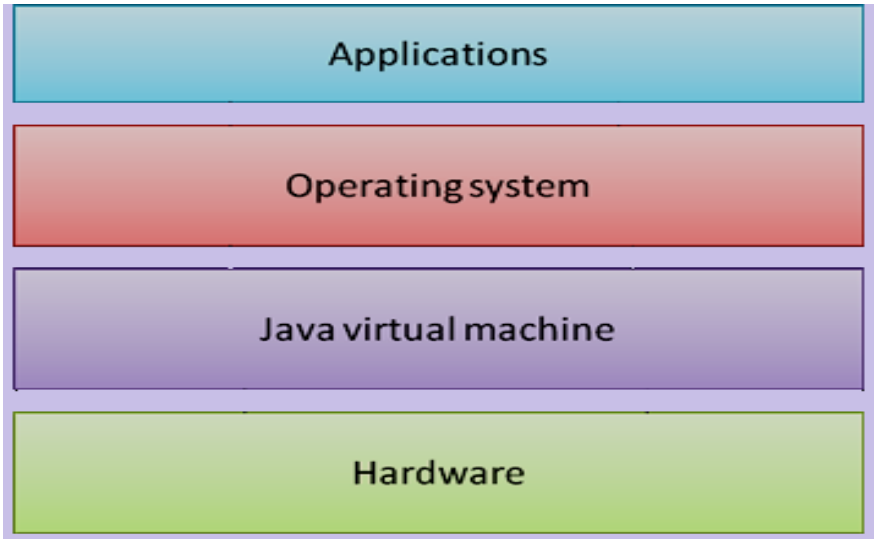

Fig. 5 Architecture of blackberry mobile OS

For Blackberry some groups of Java APIs are available. These are used to integrate with the existing Blackberry applications like phone, email, calendar, browser, and tax list and the groups are:

- User Interface APIs.
- Event listener
- Networking Data Storage APIs.
- Application integration APIs.
- Persistent Data storage APIs

In Blackberry more utilities like additional APIs for data encryption and compression, XML parsing, location based services, Bluetooth connectivity etc are available. The specific libraries of Blackberry offer extensive support of phone related features. For example it is possible to add, view or change contacts of user in the address book, manipulate call logs. Blackberry applications can be developed as standard Java ME applications extending MIDlet class defined in MIDP (Mobile Information Device Profile) specification [14]. This is opposed to Blackberry specific programs built as CLDC (Connected Limited Device Configuration) applications by extending the class UI application. Whereas MIDlet can be ported to any device featuring the used Java APIs, usages of Blackberry UI applications is restricted to the Blackberry platform and devices [14].

IV. Comparision of Different Mobile Operating Systems

Mobile Devices i.e. handheld devices have become an important part for communication purpose in human being's life. Due to change in technology and time, use of mobile devices shifted towards to Smartphones. In existing work [15]-[19], the authors basically make comparison between smartphone based operating system like Android, iOS, Symbian, Windows and Blackberry. We have presented the comparison in table 1. This

tabular presentation will help to easily differentiate among different operating systems. Some characteristics of different smartphones OS have been evaluated.

Table 2: Comparison of different mobile operating systems [15]-[19]

Parameters	Android	iOS	Symbian	Windows	Blackberry
OS Family	Linux	Darwin	RTOS	Window CE-7 Window NT-8	QNX
Vender	Open Handset Alliance, Google	Apple, Inc	Accenture on behalf of Nokia (historically Symbian Ltd. And Symbian Foundation)	Microsoft	Blackberry Ltd.
Environment (IDE)	Eclipse (Google)	XCode (Apple), Appcode	QT, Carbide.C++, Vistamax, Eclipse	Visual Studio	Eclipse, Blackberry JDE
CPU Architecture	ARM, x86, MIPS	ARM, ARM 64	ARM, x86	ARM	ARM
Source Model	Open Source and in most devices with proprietary components	Closed Source	Closed Source, Previously open source	Closed Source	Closed Source
License	Free and open source, but usually bundled with proprietary apps and drivers	Proprietary ULA except for open source components	Proprietary, Previously licensed under EPL	Proprietary	Proprietary
Written In	C, C++, Java	C, C++, Objective C, Swift	C, C++, ME, Python, Ruby, Flash Lite	C#, VB.NET, F#, C++, JScript	C, C++, HTML 5, Java script, CSS, Action script, Java
Market Share	48.8%	17.2%	0.1%	19.5%	11.1%
Market Size	Very High	High	Very low	Medium	Low
Application Store	Google Play	App Store	Nokia Ovi Store	Windows Phone Store	Blackberry World
Cross Platforming	Android supports cross platforming	iOS don't support cross platforming	Symbian supports cross plat forming	Windows support cross platforming	Blackberry don't support cross platforming
Memory Utilization	Paging, Memory Map, No Swapping	Automatic Reference Counting, No Garbage Collection	Memory management Unit and Cache resides on a SOC (System on Chip)	ROM/RAM is flash memory used for Virtual Memory storage. Programs can only run from main memory.	Contain slot for external memory supports 32GB Micro SD card at the time of full memory.
Security	Multi layer security. Most secure and usable OS	Low level software hardware and farm ware security	Gate keeper type of security every time ask for user permission	Windows OS does data encryption, leak prevention and	Blackberry provides two methods data encryption

V. Conclusion & Future Needs for Mobile OSs

With evolution of mobile phones our life is more connected and it has become almost a necessary commodity in day to day life. This is irrespective of peoples of age, class creed, color, or which part of world they belong. With the availability of powerful mobile operating system and with the tremendous growth in mobile communication technology mobile computing is projected as the future growth area in both academia and industry. There is thus vast scope of potential research and development in this area.

The aim of this paper is to reviews most popular mobile operating system features. We provide a comparison table for useful features of mobile OS that allows mobile users to make a proper choice based upon their need. We have found that android and Windows Phones are superior to others OS. Android is the best Smartphone OS in the world today because of its simplicity. We can also use it as an educational tool. Due to android as an open source operating system, the user can easily install third party applications from markets and even from unreliable sources. Due to this, it has some limitations which lead to malware attacks like virus, worms, spyware, adware and Trojan horse. Another area of research is the issue of power management for mobile phones. Increasing the battery life without increasing the weight of the phone is a challenging design issue, necessary steps should be taken to overcome these problems in for future.

References

[1] Parmjit Kaur, and Sumit Sharma. "Google Android a mobile platform: A review." Engineering and Computational Sciences (RAECS), 2014 Recent Advances in. IEEE, 2014.
[2] Michael Becher, Felix C. Freiling, Johannes Hoffmann, Thorsten Holz, Sebastian Uellenbeck, Christopher Wolf, "Mobile security catching up? revealing the nuts and bolts of the security of mobile devices.", Security and Privacy (SP), 2011 IEEE Symposium on. IEEE, 2011.
[3] "Gartner Says Android to Command Nearly Half of Worldwide Smartphone Operating System Market by Year-End 2012." [Online]. Available: http://www.gartner.com/newsroom/id/1622614/.
[4] "Number of apps available in leading app stores as of July 2014." [Online]. Available: http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/.
[5] Drago¸s Sbırlea, Michael G. Burke, Salvatore Guarnieri, Marco Pistoia, and Vivek Sarkar, "Automatic detection of inter-application permission leaks in Android applications." IBM Journal of Research and Development 57.6 (2013): 10-1.
[6] Okediran O. O., Arulogun O. T. and Ganiyu R. A, "Mobile Operating Systems and Application Development Platforms: A Survey." Int. J. Advanced Networking and Applications 6.1 (2014): 2195-2201.
[7] "ART AND DALVIK." [Online]. Available: https://source.android.com/devices/#features.
[8] CMER, (2014): "Mobile Operating System" Centre for Mobile Education and Research.
[9] Okediran O. O., Arulogun O. T. and Ganiyu R. "Mobile Operating Systems and Application Development Platforms: A Survey". Journal OF Advancement InEngineering AndTechnology July 10, 2014, Accepted: August 08, 2014, Published: August 08, 2014.
[10] Apple1, (2014): "The Application Runtime Environment." Available: http://developer.apple. com / library / ios / #documentation / iphone / conceptual/ iphoneosprogrammingguide / Runtime Environment /RuntimeEnvironment.htm.
[11] Prof. Dr. Jörg R. Mühlbacher Linz, Juni(2008), "Mobile Service Oriented Architecture in the Context of Information Retrieval".
[12] Windows, (2011): "Windows Phone 7 Platform Introduced to iPhone Application Developers", available at http:// windowsphone.interoperabilitybridges.com / articles / chapter-1-windowsphone-7-platform-introduced-to-iphone-application-developers.
[13] Xiao Feng Li, Yong Wang, Jackie Wu, Kerry Jiang, Bing Wei Liu, "Mobile OS Architecture Trends", Intel Technology Journal, Vol. 16, Issue 4, 2012.
[14] http://smallbusiness.chron.com/comparision/iphone-vs-blackberry-47579.html.
[15] Dabhi, Rajendra M., and Sunil Kumar V. Nakum. "A Paper on Latest and Upcoming Smartphone OS." International Journal 4.4 (2014).
[16] T.N.Sharma, Mahender Kr. Beniwal, Arpita Sharma, "Comparative study of different mobile operating system." Int. J. Adv. Res. Technol 2.3 (2013).

[17] Jyothy Josep, Shinto Kurian, “Mobile Os- a Comparative Study”, Journal of Engineering, Computers, Applied Science vol. 12, Issue 4, 2015.
[18] Nosrati, Masoud, Ronak Karimi, and Hojat Allah Hasanvand. "Mobile computing: principles, devices and operating systems." World Applied Programming 2.7 (2012): 399-408.
[19] Ryan Johnson, Zhaohui Wang, Angelos Stavrou, Jeff Voas, "Exposing software security and availability risks for commercial mobile devices." Reliability and Maintainability Symposium (RAMS), 2013 Proceedings-Annual. IEEE, 2013.