

ADAPTING LINUX AS MOBILE OPERATING SYSTEM

¹Kaushik Velusamy, ¹Shriram K. Vasudevan,
¹Sankari Gopalakrishnan, ²Subashri Vasudevan and ³Balachandran Arumugam

¹Department of Computer Science, Amrita University, Coimbatore, India

²Department of Software Engineer, MNC, Chennai, India

³PRIST University, India

Received 2013-05-04, Revised 2013-05-25; Accepted 2013-06-05

ABSTRACT

In this fast growing world, people are increasingly mobile; everything is fast, connected and highly secured. All these have put up the requirements on mobile devices and leads to several features being added in the mobile operating systems and its architecture. The development of the next generation software platform based on Linux for mobile phones provides enhanced user experience, power management, cloud support and openness in the design. In spite of many studies on Linux, the investigations on the challenges and benefits of reusing and adapting the Linux kernel to mobile platforms is very less. In this study, a study on architecture of the Linux, its adaptations for a mobile operating system, requirements and analysis for Linux mobile phones, comparison with android and solution technologies to satisfy the requirements for a Linux mobile operating system are analysed and discussed.

Keywords: Linux, Mobile Operating System, Adaptations

1. INTRODUCTION

In order to deliver a good quality of a software system in a short release time, many software organizations chose to reuse existing developed software systems (available code) than creating a new system from scratch. Android, an operating system for mobile devices developed by Google, reused and adapted Linux operating system. The flexibility of the Linux operating system has led to its adoption in a wide range of domains, from home PCs to supercomputers.

Many mobile organizations are moving towards Linux, which could provide high functionalities equivalent to that of the desktop computers. Recently, the deployment of Linux in mobile operating systems has risen exponentially, by different mobile phone companies like Motorola, HTC, Sony and Samsung.

Nowadays, mobile phones not only provide voice service but also multi-media message services, portable media players, high resolution touch screens, automated

voice assistant, wide range of applications and high speed data access provided by Wi-Fi and mobile broad band. Therefore, an efficient software platform is needed to support all these and the upcoming features. The evolution of the operating system starts from the PC-based OS to an embedded OS to the current smartphone-oriented OS. In all these stages, the technological advancements in the hardware, software and in networks of a mobile phone has changed from simple to intermediate to complex architectures.

In spite of limited hardware support for mobile phones such as Personal Computer Memory Card International Association (PCMCIA) cards, graphics card and internal modems, the microprocessors and the peripherals used in mobile phones are small in size but provide high processing capabilities at the same time. The hardware parts in mobile phones cannot be changed easily and are also limited in terms of disk space and CPU speed. Still the performance gap of desktop computers to mobile phones are becoming smaller and one day the mobile phones would become an exact

Corresponding Author: Kaushik Velusamy, Department of Computer Science, Amrita University, Coimbatore, India

replacement for desktop computers. Thus, they often make use of specialized hardware and hence, now, not only finding a driver becomes difficult but they are also used in changing environments. This brings the need for multiple configurations and additional security strategies.

The most important point to be observed in smartphones is, it runs on ARM core based architecture. Linux-based does not necessarily mean open source. Both the mobile operating systems the webOS and Bada are closed source platforms.

The software in the desktop operating system is mainly focused on user's productivity with the support for the peripherals that have a precise input functionality along with rich user interface, whereas the software for PDAs also allows managing personal data.

Thus the mobile operating systems cannot be self-contained, but have to be open systems. The usage of the past mobile devices had changed from just data management and local gaming to rooting and cloud related services.

In the past the mobile operating system did not have full multitasking or 3D graphics support, including accelerometers and capacitor based touch screens. Now the mobile phones are provided with increasingly sophisticated functions with rich peripherals.

The Android system built from the kernel of Linux 2.6 has also been analysed extensively. A good knowledge on the challenges and benefits of Linux adaptations would also be beneficial for practitioners interested in adapting the Linux system to new platforms.

This review aims at assessing the ease to adapt the Linux kernel into the mobile operating system and the benefits of adapting the Linux kernel instead of developing a new operating system. The rest of this study is organized as follows: Section 2 briefs about the mobile operating systems, Section 3 describes features of mobile os architecture, Section 4 presents a view on Linux as a mobile operating system and its middleware, Section 5 and 6 gives a brief description on analysis of Android OS and its architecture. Finally concludes with the adaptations made from Linux to android architecture with few discussions on some limitations and outlines avenues for future work.

1.1. Mobile Operating System

The mobile operating system is responsible for determining the functions and features such as Internet Access: browsers for Compact or Full HTML, HTML5

AP(aWML1.x (wireless mark-up language) and XHTML), Adhoc-Hotspot, hardware accelerated graphics, 3G connectivity and location-sensing. Messaging which includes e-mail, short message service, enhanced message service, multimedia message service and multimedia-display of still image files, playback of animation, movie, audio files and Applications-personal information management, games, electronic commerce, novel applications to utilize the rich peripherals. Program Execution Engines: Java runtime environment, WMLScript, ECMAScript. The above functionalities have become similar to those of the PC. Today, all these features are standard. Projector, face and gesture recognition, near field communication, augmented reality and holograms are still in the developmental stage. As a result, a more sophisticated OS is required to carry these along with the upcoming functionalities and Linux is one of those. Requirements of all such multi functionalities and the quality of mobile phones leads to the complexity of software structure in mobile phones.

The architecture of MultiNets demonstrates the methodology to perform switching in Linux based mobile Operating systems such as Android which provides a smart phone platform to save energy, offload data traffic and achieve higher throughput. Analysis shows that mobile data traces collected from real users with real-time switching can save up to 27.4% of the energy and offload of up to 79.82% of the data traffic. It also achieves 7 times more throughput on average.

1.2. Mobile OS Architecture

Current mobile phones utilize a real-time OS for embedded systems which includes drivers for peripheral devices, communication software, libraries shared between applications and application software are placed on top of that real-time OS (**Table 1**). In recent years, various innovations for more efficient software development have been achieved. Many sophisticated OSs such as Linux for PCs and Symbian OS for mobile equipment with a large number of software development environments are available and enormous types of application software are distributed in the market.

These sophisticated mobile operating systems provide virtual memory spaces and the memory protection mechanism by which the application programs are protected. By this mechanism illegal memory access during the software development can be easily detected.

Table 1. History and development of mobile phones

1973	First mobile phone device by Motorola
1978	Advanced Mobile phone system (1G)
1990	GMS standard (2G)
1993	First smartphone by IBM with touch screen
1996	Windows CE Handheld devices.
1998	Psioncompany develops Symbian
1999	First Nokia phone with Symbian S40 (7110)
2002	First smartphone by Blackberry
2002	3G telecommunication technology
2007	Apple I phone with ios introduced
2008	Android 1.0 with HTC dream
2010	Microsoft launches Windows Phone OS
2011	MeeGo first Linux mobile by Nokia and Intel
2013	The Ubuntu phone *

However there are few requirements for the mobile phones which distinguishes them from the personal computers:

- The first and the foremost factor is the CPU power. Mobile phones are driven by secondary batteries which is much lower than the wired power supply in Pcs
- CPU frequency of the current mobile phones is between 50 MHz and 100 MHz while a PC frequency is between 2 and 3 G Hz
- Memory size of the current mobile phones is between 16MB and 32MB while that of PCs is 512 GB-1 TB. As far as now mobile phones do not have hard disks
- Mobile phone's primary purpose of notification for call should not be affected even when the user is playing music or surfing the internet
- Mobile phones, being an extension of wired-phones there shouldn't be any incoherency in voice communication

1.3. Power and Memory

In order to increase the utilization time, efficient power usage is required. Advanced Configuration and Power Interface (ACPI) provides power management such as suspend/resume mechanism and device configuration by the operating system.

Yua *et al.* (2010) the two graphs (**Fig. 1 and 2**) show the total energy consumption of mobile phone manufacturing, usage and network, out of which usage is taken for the study on the power consumption in mobile phones. The second graph shows the energy consumption vs different functionalities of mobile phones during its entire life cycle. Along with the minimal clock rate, the kernel threads should also be kept suspended when the system is idle.

Usually the Linux kernel is stored in a compressed form in the ROM or file systems in disks (if any), on booting its memory images are expanded in the RAM instructions in it are executed. Since RAM is one of the cost factors in mobile phone, reducing the size of RAM is always required.

In order to avoid large RAM usage, codes in Linux kernel, its applications and libraries should be directly executed in its ROM area without being copied and decompressing the kernel into the RAM area. This in place execution mechanism increases low speed ROM area and saves high speed RAM area. There should also be a reduction in system-wide approaches. Memory reduction efforts are applied not only to the kernel, but also to the middleware and the applications.

In this process of reduction, the Linux kernel and libraries should be customized and memory alignment used in various kernels, should be reconfigured. Thus the Memory Management Unit (MMU) features are highly needed for the mobile application.

Example

uCLinux, which is suitable for embedded systems, merged into Linux 2.6, is not appropriate for mobile phones because it does not have MMU features.

During booting, the compressed kernel file is copied on the disk, uncompressing the copy, starting up daemon (the background processes) and starting up the middleware (middleware is computer software that provides services to software applications beyond those available from the operating system).

During this process, Linux in a PC takes several minutes to set up device drivers and check consistency of file systems while in mobile phones, it's booted up within less than a second.

Deferred initialization of device drivers is one of the effective solutions for fast booting processes in mobile phones.

1.4. Interrupt and Scheduler

The real time response in Linux is degraded by the long pre-emptive process. The interrupt mechanism facilitates this and latency is found to be in milliseconds for PCs and in microseconds for mobile phones.

The time from when an external interrupt arrives and the process pre-emption occurs, the interrupt handler is executed. All the execution in this interrupt context in the corresponding device driver is executed in the kernel. This sends the data to the corresponding user process. Finally the process is pre-empted. This leads to a delay in the interrupt process.

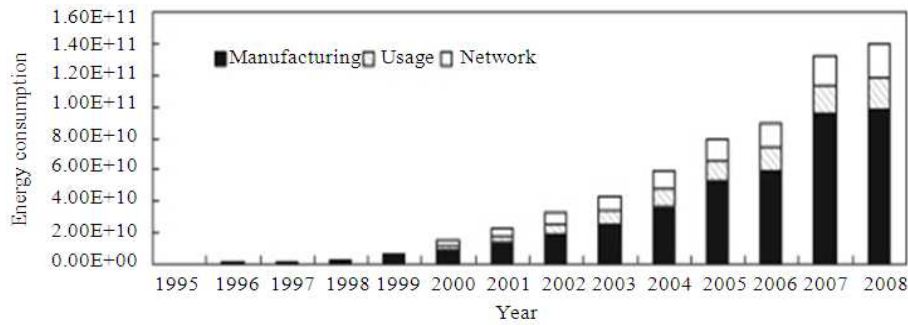


Fig. 1. Total energy consumption of mobile phone manufacturing, usage and network infrastructure

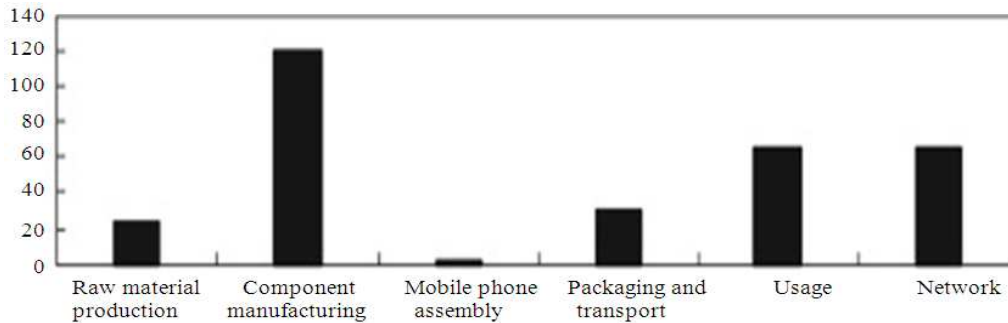


Fig. 2. Energy consumption of mobile phones during its life cycle

In UNIX, the interrupt execution is serialized. In the traditional Linux, critical regions are guarded from other process access by prohibiting process pre-emption during a system call execution.

To improve the delay, the interrupt is executed concurrently. A tasklet and Symmetric Multi-Processor (SMP) kernel-spinlocks is a solution.

For this enhanced Linux kernel an improved scheduler to shorten overhead of process pre-emption is needed. To schedule process in a constant time, a kernel scheduling design called O(1) scheduler is used, reducing overheads and deviation, by providing execution times with fixed upper bound.

Laxmikantha *et al.* (2013) besides modifying the kernel, another approach is to Integrate Linux with real-time OS (Fig. 3). The Linux kernel is implemented on the top of real-time OS. Real-time processing is done in real-time tasks and not as execution units in Linux.

Real-time enhanced Linux and the integration of Linux and real-time OS is based on ‘2-CPU solution’ for mobile phones. An Application CPU (A-CPU) for mobile application services, its real time enhanced Linux and a Communication CPU (C-CPU) for communication processes.

1.5. The File System

The file system is checked for inconsistencies in Linux and normal reactivation during system restart and sudden power off. This requirement is achieved in Flash file system which is a file system particularly designed to store files on devices with flash memory. As the number of mobile devices increases, the cost per memory size decreases and the capacity increases. The disk file system can be used on a flash device. Similar to the disk drive, the flash system are used on the flash devices for several reasons:

- Erasing blocks: The memory blocks in the flash have to be erased explicitly every time before they are written. This significant time taken to erase the unused blocks is beneficial while the device is idle
- Random access: Disk file system has high cost disk seek time whereas flash memory devices do not impose any seek latency
- Wear levelling: Repeated overwriting of single block tends to wear out the flash memory devices. Hence they are designed to spread out writes evenly

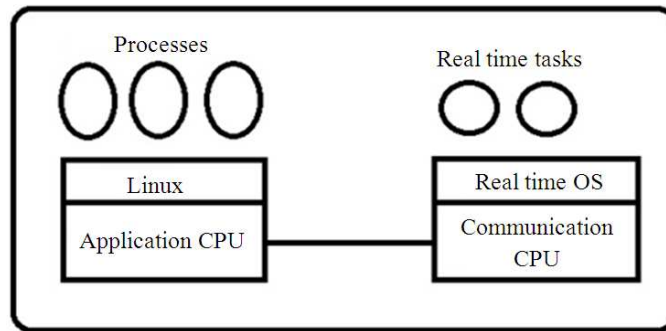


Fig. 3. Interrupt processing in real time linux

A file system designed for high write throughput where the updates are written sequentially is a log-structured file system. Such a file system which is used to increase reliability during power down and reset has all the desirable properties such as Journaling Flash File System (JFFS) and Yet Another Flash File System (YAFFS) which are included in the Linux kernel from 2.4.10.

Flash memory is best used with a controller, else a specifically designed flash file system must be used to hold long erase times of NAND flash blocks which are mainly used in embedded flash memories. Removable flash memory cards and USB flash drives have built-in controllers with dedicated algorithms to solve wear levelling, bad block recovery, recovery from power loss, garbage collection and error correction. Whenever the flash store is updated, a new copy of the change will be written over to a fresh block, pointers remapped to it and the old block will be erased. The device file which acts as an interface for the device driver, appears in the file system as an ordinary file, helps to interact with the device driver using standard IO system calls.

1.6. The Security

Jones (2008) to implement security CPU, memory and OS resources must be guaranteed to be allocated in specific duration. Security functionalities in the kernel are enhanced with Linux Security Module (LSM) which provides a framework to protect the mobile phone software from attacks with various security policies. They are provided along with the system call executions in which security checks, provided in a separate security module are performed. SELinux3 is a set of modifications applied to Unix-like operating systems to provide security policies which is a security enhanced Linux module on the LSM framework.

Figure 4 Security-Enhanced Linux (SELinux) is a Linux feature and not a Linux distribution, which came

into use from Linux Kernel 2.6. It is a set of modifications made to the kernel space and the user space, so that it can be added to various distributions of Linux. It provides a supporting mechanism for access control security policies with the help of Linux Security Modules (LSM). Its architecture separates enforcement of security decisions from the security policy and filters the software charged with security policy enforcement.

Rao *et al.* (2010) a malicious program can expand the thread stack and destroy other thread stack area. In order to protect this thread violation, the thread guarded page is allocated on the top of the stack and the threads are implemented in the kernel like Linux 2.6. Thus whenever a read/write operation is performed in a thread guarded page, it is blocked and the thread stack violation is detected.

In the case of execution of a malicious application, access to personal information such as phone directories should be handled securely so that the data is not altered and not sent to the outside world. Access to critical information should also be restricted.

1.7. Using Linux AS Mobile OS

Linux supports various installation methods, works in heterogeneous environments and needs smaller resources. Thus it is best suited for mobile devices.

Recently, the deployment of Linux in mobile operating systems has rapidly risen which is due to lesser cost in adapting existing software than building a new one.

The flexibility of the Linux operating system has led to its adoption in a wide range of domains from home PCs to supercomputers. In the evolution of the Linux kernel, the growth rate is found to be super-linear and a decrease in average complexity. Most part of the code in Linux consists of device drivers, which are relatively independent of each other.

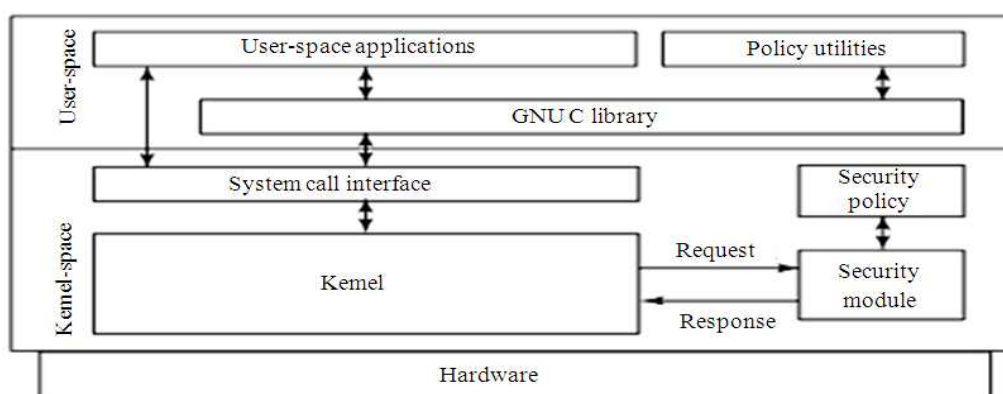


Fig. 4. Framework of SE- Linux enforcing security policies

The disadvantages include a comparatively large memory footprint, complexities of user mode and kernel mode memory access a complex device drivers framework. Application program development environment helps programmers to develop program in PCs and port it into mobile phones with a recompile.

Even if the OS in mobile phones is Linux, the header files, library files and system call APIs are different from those in a PC. The middleware, libraries and header files are implemented as the guest Linux and are independent of the host Linux. User Mode Linux (UML) is a Linux kernel implementation in a user space which can provide application program environments, independent of the host Linux and helps in the implementation of the guest Linux.

Development support environments are required to develop optimized Linux kernel for a target product.

1.8. Middleware

Middleware is a logical software layer beyond those available from the OS, located between the kernel and application software, implemented in the protected memory space. The middleware are usually supplied from third party vendors or open source software development communities. Mer, a mobile-oriented software distribution targets hardware vendors to serve Linux-based operating systems. Graphical Mer interface also includes window systems, Multimedia functions and Network communication protocol stacks.

The primary purpose of a mobile phone to catch a call should be maintained. When a call comes, the interrupt management mechanism interrupts the current

applications and notifies the user with a ring tone, vibrating the phone, or a graphical icon.

APIs are needed to install third party application software easily. X window system and GTK toolkit occupy large amount of RAM area. Reduced RAM consumption, double buffer extension to draw image data quickly, font sets with small size and quicker loaded time are required for the efficient use of XWS and GTK. In Section 5, the adaptation of the Linux kernel into Android is investigated.

1.9. Android-an Analysis

Khomh *et al.* (2012) though the intention of this study is not to use Android, a good understanding of the challenges and benefits of Linux adaptations to mobile phones becomes inevitable without its study. Android, the term which literally means Andro-humanoid robot is not Linux (but a Linux-based OS with enhanced kernel portion of Linux), designed primarily for smartphones and tablet computers, with GNU utility packages ported to ARM which takes a role in the Abstraction layer between Hardware and Software without anative windowing system. Google has adapted one of the most reused computer operating systems (i.e., Linux) into a mobile operating system (i.e., android).

Android adapts the existing Linux Kernel v2.6. It was originally founded by Android Inc., in 2003, later purchased and developed by Google since 2005. Android is open source and Google released the code under the Apache License. Android, the low-cost, customizable, lightweight operating system has a large community of developers thus extending its

functionality and making it a widely used smart phone OS overtaking windows and Symbian. Android can run on processors ranging from 600 MHz single core to any Quad core powerful 2 GHz processor. Android OS are also used in wrist watches, cameras televisions, embedded devices-home appliances and project glass-Google glass. The Ouya is a video game console running its own version of the Android OS. The android applications are written primarily in a customized version of Java. Android SDK which includes debugger, libraries and device emulator, Java JDK, Eclipse Android Development Tools plugin are needed for development. Android Debug Bridge (ADB) is a versatile command line tool that lets you communicate with an emulator instance or connected Android-powered device. It is a client-server program that includes three components which can also be invoked from the shell using adb command.

The adaption involved only a very small percentage of files from the Linux kernel to be modified. The merging of changes on files affected only a minor part of the android files. Most of the bugs experienced by users of the Android kernel are fixed in the Linux kernel repository itself.

1.10. Android-Architecture

The minimal requirement features for android includes ARM-based chipset with a minimum memory of 128 MB RAM; 256 MB Flash External with mini or micro storage. Primary display of QVGA TFT LCD or larger, 16-bit colour or better with 5-way navigation with 5 application keys, power, camera and volume controls and Standard mini-B USB interface with Bluetooth or Wi-Fi support.

Figure 5 Binder Driver is a specific android inter process communication mechanism between applications and a remote method invocation mechanism with synchronous calls between processes. The android process can call a routine in another android process using binder to identify the method to invoke and the arguments. It maps the process memory address to kernel address. The binder implementation is in the kernel source at: Drivers/misc/binder.c, with include file: include/linux/binder.h

Different wake locks features in android provide different power schemes. The Native Libraries in android includes C/C++ libraries with Custom libc,

Bionic. WebKit is used as Web Browser kernel. Packet Video and OpenCORE platform provides the Media Framework and database provided by SQLite and Light weight SQL.

Figure 7 Android Runtime includes core Libraries and Core APIs for Java with Data structures, Utilities, File/Network access and graphics. Every Android application runs in its own process, with its own instance of the Dalvik virtual machine so that the device can run multiple virtual machines efficiently. The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

Figure 6 The .java file is converted to .class file using javac. The .class file is converted to Dalvik Executable. (dex) format which is optimized for minimal memory footprint using dex which is finally converted into .apk (android application package file) using aapk. The .apk file is run using adb which communicates with an emulator instance or connected Android-powered device.

Table 2 Application Framework in android includes the Core platform services and Activity, Package, Window, Resource Managers. It also provides services to the hardware and access to lower-level API through location Manager. Telephony, Bluetooth, Wi-Fi, USB, Sensor Service also works in this framework.

The software adaptation includes three principal phases. The initial modification of the existing system required to reuse its functionalities in the adapted system. The maintenance of the adapted system to keep it updated with changes in linux. Fix bugs carried down from the original system as well as the new bugs introduced during the previous development phases.

In the process of adapting the Linux kernel into Android, it is found that 99% of the functionalities of Linux kernel 2.6 were reused and only 0.7% of those were modified. Analysing the functional similarity between the two systems and the amount of changes performed to integrate Linux functionalities into Android, it is found that 84% similarities exist between the source code of Android 1.1 and Linux kernel 2.6 and only the file system and memory management modules have been modified. Finally, the results of this study can be of interest to researchers and developers who are interested in adapting any large legacy systems into new platforms.

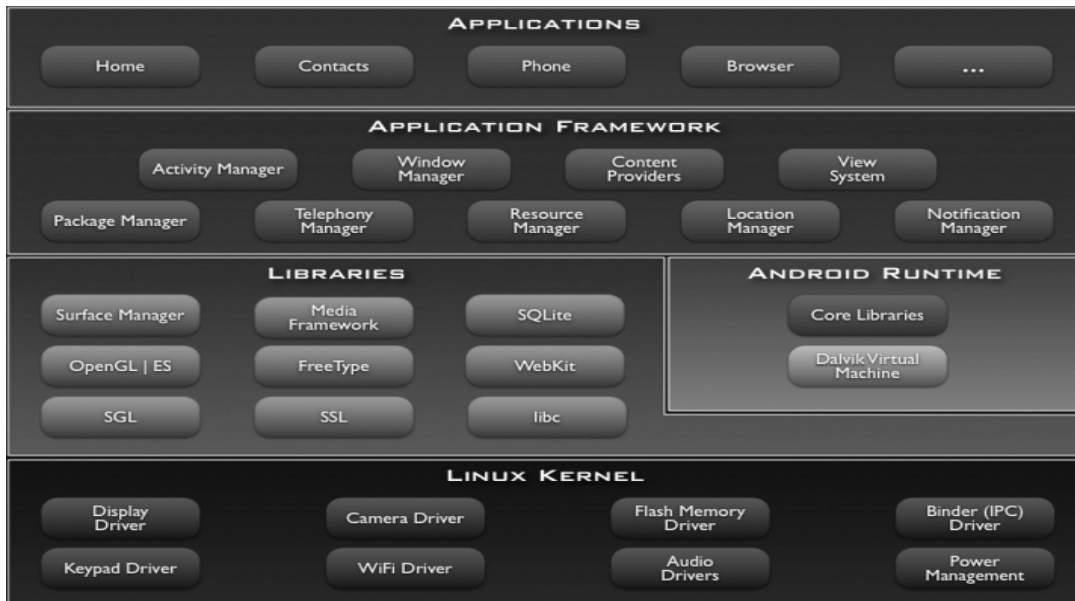


Fig. 5. Android-architecture

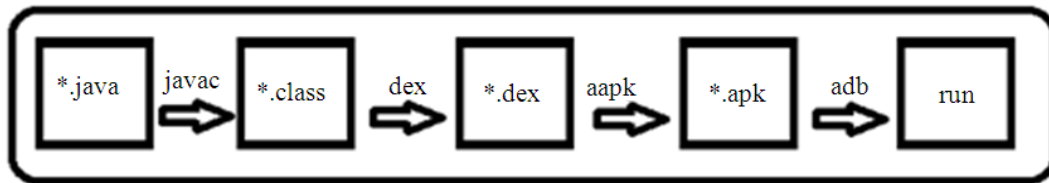


Fig. 6. File conversions in Dalvik virtual machine



Fig. 7. Adaption of Linux kernel by Android with added additional features

Table 2. Application framework features and roles in android architecture

Feature	Role
View system	Used to build an application, including lists, grids, text boxes, buttons and embedded web browser
Content provider	Enabling applications to access data from other application or to share their own data
Resource manager	Providing access to non-code resources (localized strings, graphics and layout files)
Notification manager	Enabling all applications to display customer alerts in the status bar
Activity manager	Managing the lifecycle of applications and providing a common navigation backstack

2. CONCLUSION

Through a rigorous study made for the review on Linux, many things are unwounded. Starting from analysis of what a Linux OS is, how it is constructed have been touched in an elaborate manner. Then the gear is shifted towards adapting Linux as a mobile operating system. Starting with understanding Android, its architecture, its novelty and many aspects of its construction and utilization, the study moves further to let the practitioners know how to adapt the Linux for the handheld smart devices. To add as a tail piece, clear cut views and steps are also reviewed which will be an icing over the cake for the reader.

3. ACKNOWLEDGEMENT

We thank Amrita University for providing us continuous support and guidance. Most importantly we thank Computer Science and IT departments for providing us motivation.

4. REFERENCES

- Jones, M.T., 2008. Anatomy of Security-Enhanced Linux (SELinux): Architecture and implementation. DeveloperWorks@.
- Khomh, F., H. Yuan and Y. Zou, 2012. Adapting Linux for mobile platforms: An empirical study of Android. Proceedings of the IEEE International Conference on Software Maintenance, Sept. 23-28, IEEE Xplore Press, Trento, pp: 629-632. DOI: 10.1109/ICSM.2012.6405339
- Laxmikantha. K, Kishore M and H M Guruprasad, 2013. Real-time Linux using RTAI. Int. J. Comput. Appli. 61: 31-33. DOI: 10.5120/9894-4461
- Rao, K.K., S. Nagaraj, G.A. Rao, G. Anuradha and GSVP. Raju, 2010. Designing a thread migration facility for linux operating system. Int. J. Comput. Appli., 1: 103-108. DOI: 10.5120/493-806
- Yua, J., E. Williamsb and M. Jua, 2010. Analysis of material and energy consumption of mobile phones in China. Energy Policy, 38: 4135-4141. DOI: 10.1016/j.enpol.2010.03.041