iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VIII **Month of publication:** August 2020

DOI: https://doi.org/10.22214/ijraset.2020.30932

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VIII Aug 2020- Available at www.ijraset.com

Developing Custom ROM based on Android using AOSP

Shreyas S[1], T Tejo Veena Sri[2], Poshith N[3], Shesha Sai Charan R[4], Ravi Kumar V[5]
[1, 2, 3, 4, 5]*Department of Computer Science and Engineering, Vidyavardhaka College of Engineering, Mysore, Karnataka, India*

Abstract: Android is an OS by Google based off multiple platform programming languages and is the most extensively used mobile OS after 2013. Apart from being OS for mobile device, it can also be used for other devices. It also provides set of platform software stack enhancing usability of devices. Started with Android Alpha, and the latest public version is Android 10. Android, being an open source allows any user to tweak the experience to their preference by modifying its source code and improves its performance. Exploring Android allows the user to know the inner workings of the system and openness of it in-turn helping to understand the various coding decisions and multiple opportunities for creativity. The main intent of this research is to build the custom ROM utilizing Android Open Source Project(AOSP) which enhances the performance and re-designed UI with customizable option to the user.

I. INTRODUCTION

Android, developed by Google, is an open-source OS which runs on smart phones. The figure 1.1 shows the pie chart of various operating system available. It is based on Linux Kernel.

Its open nature provides freedom for developers to fulfil increasing demand of user needs. Statistics show that Android covers nearly 80% of smart phones.

The figure 1.2 shows the number of application available on Google app store. Android keeps the background processes alive unless and until other process require memory. Android categorizes apps into different ways like Visible , Content provided , Empty , Foreground , Hidden , Secondary server and its all managed by LMK (Low Memory Killer).

A Custom ROM is a standalone variant of Android that includes Kernel, apps, and all other services that is required to run the device. Setting up Android build system and ready to compile on a Linux machine to make changes and compile [1]. Involving installing required tools such as git, flex, tool chains and other dependencies for the build system further making changes to the Android Open Source Project (AOSP) [1] forked/clone directly from Google's git repository (git.googe.com)to get generally out of said android device with respect to performance and UX side enhancements(New features and alterations for existing features)[1].

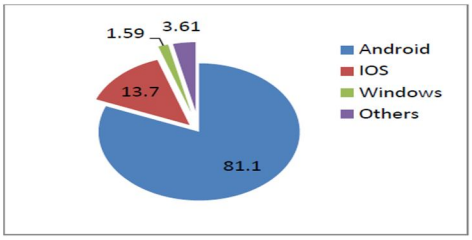


Fig 1.1 Number of Mobile OS users

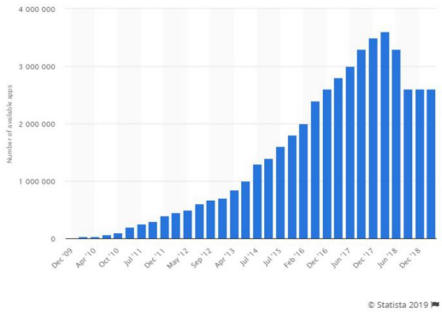


Fig 1.2 Number of Android applications available

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VIII Aug 2020- Available at www.ijraset.com

A. Android Architecture

Its architecture consists of six subdivisions as shown in figure 1.1(a):

1) *Stock Applications:* Android has some basic applications by which the developers can easily make use of that functionality instead of building such functionality. Examples of those applications are Phone , message , Email , Clock , Contacts , Web Browser etc.
2) *API Framework:* It's the whole list of capabilities of Android and it is composed JAVA. These API considered the core API that will be re utilized at the hour of formation of the application [1]. The abbreviation for API is Application Interface and they include
3) *Resource Manager:* Resources are a set of supplementary files and static contents that our program uses, such as bitmaps, layout definitions, user interface strings, animation instructions, and more[1]. One should always externalize some of the application resources such as images and strings from the code, so that we can maintain them independently[1].
4) *Activities:* Activities act as the entry point for interaction of user with an app, and are one of the core elements of Android platform apps. They act as central to how a user navigates internal to an app (as with the Back button) or between apps (as with the Recents button).
5) *Notification Manager:* Any message that is displayed outside application is known as notification. User Interface is used to notify the user with reminders, messages from other devices, or other information related to your app. By tapping on notification user can open it otherwise they can directly visit application[2].
6) *Content Provider:* Content providers help an application manage to access the data stored by itself or stored by other applications, and it makes a way to share those data with other applications. They increase security of the data by encapsulating the data. Content providers links data in one process with code running in another process [2]. Implementation of the content provider has many pros [2]. The most important one is that you can configure a content provider to allow other applications to securely access and modify your app data
7) *View System:* The View objects are usually called "widgets" and can be one of many subclasses, such as Button or Textview [2]. The ViewGroup objects are generally called "layouts", facilitates a different layout structure, such as LinearLayout or Constraint Layout.
8) *C/C++ Libraries:* Some of the core Android Components such as HAL are created using library functions written in C or C++. Through utilization of API framework you can access functionalities of these libraries. To access native libraries Google provided NDK which stands for native development kit[2]. Some C/C++ Libraries are OpenGL | Es , SSL , SQlite etc.
9) *Android Runtime:* ART is software layer between applications and operating system. It offers a mechanism to execute java language. ART mainly accomplishes two major activities:

a) Executes Android framework and its Applications applying JIT, hybrid model of Interpreter and Profile based AOT[2].

b) Memory Management utilizing Memory allocator and Concurrent compacting Garbage accumulator.

10) *ART:* It is utilized by the Android operating system. Replacing its predecessor Dalvik, ART converts the application's bytecode instructions into local instructions that are later executed by the device's runtime environment. ART and Dalvik are compatible runtimes executing Dex bytecode, so that the applications worked by executing with Dalvik should also work when executing with ART [3]. Anyhow, some methods that work on Dalvik donot work on ART.
11) *Hardware Abstraction Layer:* The Hardware Abstraction Layer is defined as a standard interface for hardware vendors to implement, which helps Android to be agnostic towards lower-level driver implementations. Usage of the Hardware Abstraction Layer allows user to implement the functionalities without affecting or modifying the higher level system[3].
12) *Linux Kernel:* Linux kernel is the largest collaborative software project ever. In 2016, more than 4,000 distinct developers from over 450 distinct companies contributed to the project and there were 6 releases, each comprising between 12,000 and 16,000 several changes. By the end of 2016, the size of the Linux kernel was slightly more than 56 thousand files, comprising of 22 million lines of code, build scripts, and documentation. The kernel variants numbering began at 2.6.x, where x denotes an incrementing number that altered on every release. Since then, the kernel version has now moved to 4.x that led to 2 major version changes. These version numbers are selected by the mainta Fig 1.1(a) Android architecture used by higher minor release numbers [3].

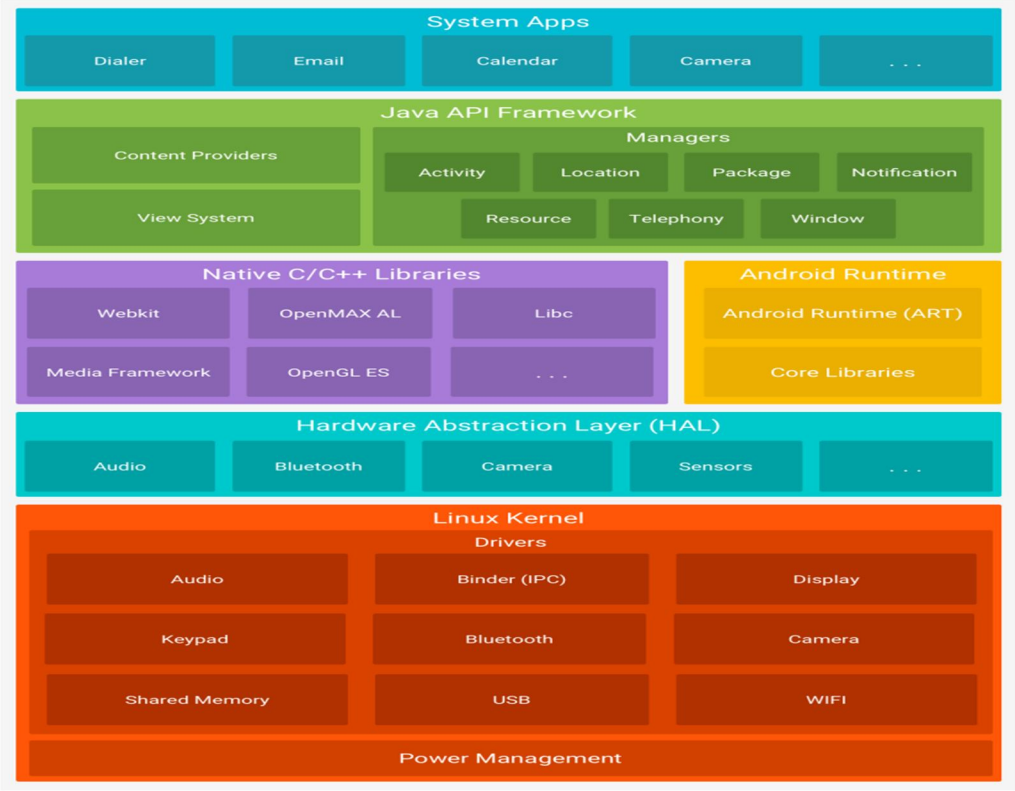


Fig 1.1(b) Mi A1 Device

Device Specifications
Display - 13.9cm (5.5)FHD display
Camera-2x Optical zoom Dual camera
Full metal body
RAM-4GB High-speed RAM
Massive 64GB storage
Battery -3080mAh(typ) battery
Processor-Qualcom Snapdragon 256
Developer name –tissot
Seamless Update-A/B system upgrades, also called as seamless updates, guarantees an operational booting system remains on the disk during an over-the-air (OTA) update. This technique minimizes the probability of an inactive device after an upgrade.

II. RELATED WORKS

A. Chirag Kanthed, Yagyapal Yadav [1] have discussed about the building custom ROM and the difficulties arising from new features of Android like RAM usage, memory usage. They explained how the RAM availability can be increased, memory can be restricted and delay free interfaces can be developed.

B. Charan K.V, S. P Sharmila, A. S Manjunath have [2] explained several constraints in migrating Android to different embedded devices other than mobile devices. They presented the ways to build custom ROM in local environment and also preparing Linux-kernel for different platform which was their success. They explained how it can be transported to other devices like cameras, desktop etc.

C. Abeer Aljarrah and Mohamed Shehab [3] proposed two different approaches inorder to enable secure apps for possible floating malicious windows. One approach is for application level to detect the floating window and the other approach is for system level which not only detects that malicious floating window but also include an event handler when floating windows are displayed on secure regions of apps. They also evaluated their efficiencies.

D. Kumar Vimal & Aditya Trevedi [4] have tried to propose a memory management scheme for Android to decide which applications can be eliminated from main memory and also tried to increase the response time of application. The proper device setting helps in decreasing the applications response time.

E. Saurabh Manjrekar and Ramesh Bhati [5] tried to look into the view behind the Android wheels and mainly focused on custom ROM. Modifying official ROM results into customised ROM. They also explained how custom ROM can be installed along with pros and cons and rooting method through which the full control over the Android system can be achieved since it involves gaining access to root.

F. Parikshit Rajput, Vinay Koraganti, Biswajeet Champaty [6] have explained how custom ROM is better than stock ROM and how they had customized the ROM for Lenovo A7000 device.

III. PROPOSED METHODOLOGY AND DISCUSSION

Android is a widely used after it is acquired by Google is growing day by day. It is mainly used in mobile devices such as smartphones, tablets, smart watches and so on. After its been released it always came up with new features, more storage, more RAM etc. but along with the new functionality the number of problems also increasing day by day.

1) *RAM:* To avoid the more use of RAM in low memory devices is always required since it is to be used by many of the processors. As per the need user installs many different applications in the system but due to low RAM some problems may occur, system will reduce its performance. For that RAM usage must be optimized and that's what we actually did we modified Low Memory Killer Algorithm so that invisible application must be killed and also carefully use if and else statement and end these statement to avoid memory leakage. We also reduced the amount of extra cache that was allocated every time to reduce the memory usage we also preloaded dex cache in Dalvik also validated Just in Time –off option has introduced ArrayMap and is widely utilized in framework as a lighter-footprint replacement for HashMap so that RAM availability will increase. Added Activity Manager[4]. Is Lower Ram Device () is to enable apps to identify when running on low-memory devices and decides to disable large-RAM characteristics.

2) *Memory:* Most of the Android devices comes with limited memory in the system and some of the memory is been used by the operating system as we are building custom ROM the memory requirement for that custom ROM is almost less than half then that of the stock ROM and rest of memory will not be used so we proposed a memory scheme so that about 1GB memory will be utilized to be used as per user demand[4] by providing this we can increase the amount storage that is available to the user hence user is free to install any application required without memory shortage.

3) *Device Tree:* A device tree is a data structure which illustrates a particular computer's hardware components including memory, CPU's, buses and its peripherals that are utilised and managed by the kernel of an operating system. Instead of hard coding the machine details by an operating system, it mainly describes the hardware that is readable by an operating system. A device tree is a tree or a cyclic graph with named nodes the node name is a label to identify the node. A node may contain various properties arranged in name value format[4]. The name comprises of a string, while value can be an array of strings, bytes, numbers, or a mixture of types. We have written the code from scratch so that no inbuilt tree function is used which helps in optimizing of the device to perform more task easily and also done code cleanup for some templates in the tree that were pre requisites so reducing the number of lines to some extents we have optimized the code to work for android 10 also added various camera parameters for better performance in this we have mentioned the entire device architecture on how and where to take the functions from where the inbuilt function is utilized and where the custom built function is utilized[4].

4) *Vendor:* The Vendor Native Development Kit (VNDK) is a set of libraries exclusively for vendors to implement their hardware abstraction layer. The VNDK ships in system image and is dynamically linked to vendor code at runtime. Android 8.0 and its next versions allows framework-only updates in which the recent versions can be updated with the system partitions while the vendor partitions are left unaltered[5]. This refers that binaries built at several times must be able to operate with each other; VNDK covers API/ABI changes across Android releases. We have added various vendor blobs to make the device work faster such as camera related drivers and linked the parameters to device tree we have optimized the blobs to support for android 10 with optimized speed[5].
5) *Battery Drain:* In Stock ROM there are lots of API and different application which are not usually required by the user but it is there in the system and that leads to battery drain. In building custom ROM we will not add those apps which is not required because if we require those apps we can install them from the Android Market where Thousands of applications are available[4].
6) *Interface:* In stock OS the user can notice drop frames while using most of the resources but in custom ROM we will provide a lag free interface and also provide user with lot of customization where he/she could utilize best of the device user will be free to choose from various different battery styles even could customize with navigation bar and also provide a battery bar at the top of the status bar[5].

IV. RESULT AND ANALYSIS

In the table 4.1 we analyse all the features that are available and also uses of those features. We also see what these features will do and why are they useful.

Table 4.1 Available features and their corresponding actions

Features	Actions
Brightness	1.Able to adjust brightness Add a + and – icons to adjust brightness level 2.users can even adjust location of brightness slider even at top or bottom
Network Traffic	User will be able to see the upload and download speed without any third party application.
Pulse navigation bar	This feature brings navigation bar into life when the user plays any music which builds an interest to use it.
Battery bar	This feature adds a bar top of the device where the user can get a glance on the battery status without utilizing space.
Recent Notification	User is able to get a glance of the latest notification when he unlock the device so no need to just slide the notification panel.
Fingerprint scanner	The user can use his fingerprint scanner to access notification panel and disable it by just a swipe.
Battery Icon	User is free to change the battery icon from a limited image .So that it brings a change in the user perspective.
Date and Time	User is free to change the location of the date and time either left right or in the middle of the top screen and give more room for notification to appear.
Custom carrier label	User can use his or her name as the carrier name instead of actually showing bsnlairtel all the time.
Gaming mode	User can enable this feature whenever he is gaming by which the device boosts up it performance by overclocking the processor and providing more attention towards game.
Custom LED notification	The user can customize his LED light to show different colour based on notification it receives ex: Low battery (RED light) Normal message (Green light)

The table 4.2 shows the comparison of custom ROM over different operating systems like iOS, blackberry based on few characteristics.

Table 4.2 Comparison of Custom ROM over different operating systems

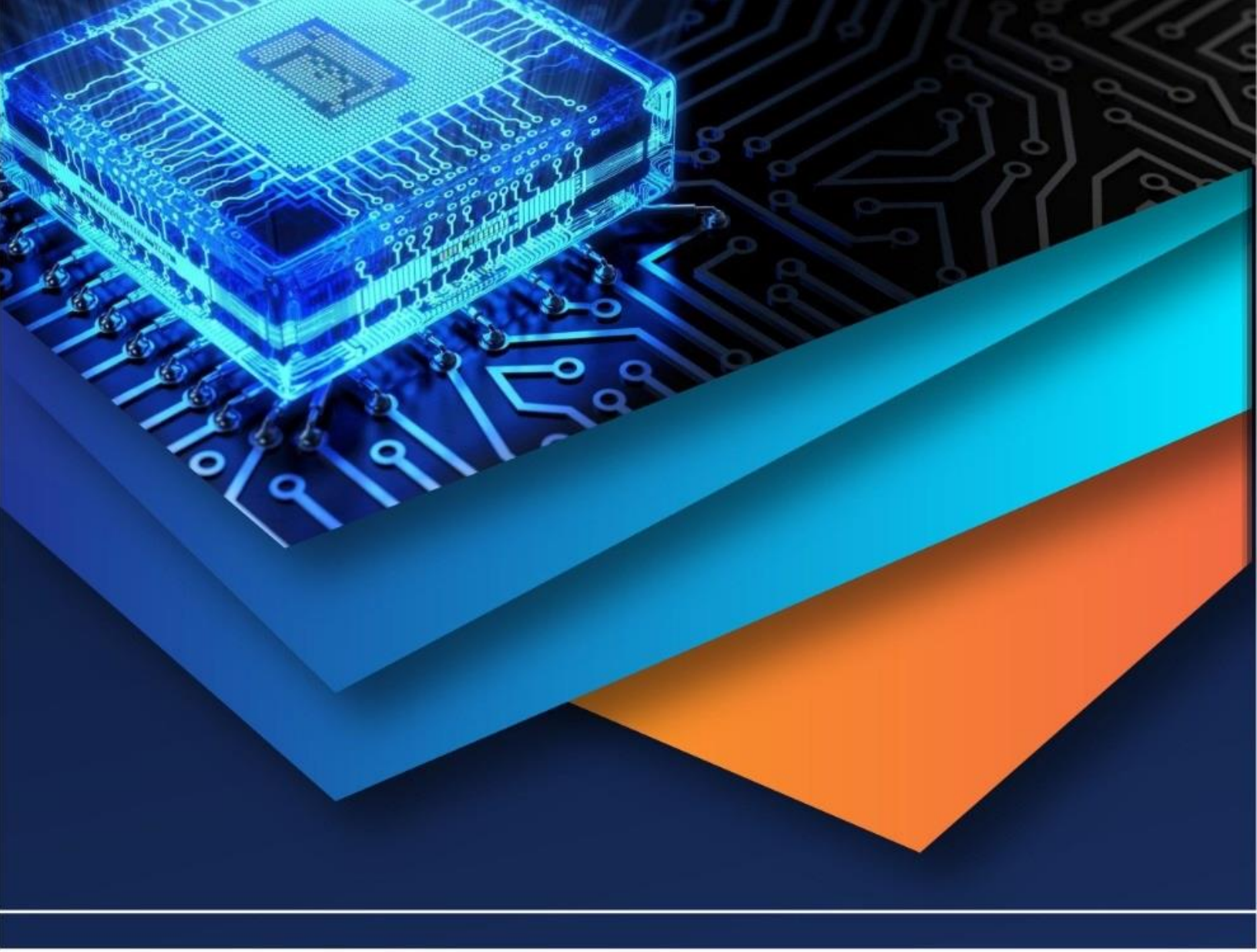
Parameters	(Memory Management, Security and Performance) Comparison		
	Custom ROM based on Android	iOS	BlackBerry
Memory Management			
Memory usage	High	Low	Low
Memory used for App Handling	RAM	RAM	RAM
Process running in Background	Efficiently	Efficiently	Not Efficiently
Use of Garbage Collector	Yes	No	No
Background Processes	Do not freeze	Freeze	Suspend
To increase process Speed	Uses internal memory	Don't use internal Memory	Uses internal
Interface	User Friendly	User Friendly	Not User Friendly
Increase in Memory Demand	No 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	Yes	No	Yes
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
Customizability	Allowed	Not allowed	Not allowed

V. CONCLUSION

Building an operating system by Android source is quite challenging and it is really complex. This research determines that all the process we implement to build and customize the ROM will result in optimized RAM and also will result in low memory usage for operating system[6]. Improved RAM usage will result in better application response and good operating system response and more memory will give access to store more where normally will not be able to store any file we have also added the support cam2api for enhanced camera performance also made code clean-up for unwanted parameters which served no purpose and added some vendor blobs to optimize the speed. We have also added some major customization to the user end so the user need to stick with just one kind of feature and can customize based on the option provided. Utilization of custom ROMs, operating system and kernels made by experienced developers help users to utilize their phone's complete power. Generally, Custom ROMs allows for better understanding of aspects and features that can be added or available to our phones. The most defining aspect of custom ROM development is the fact that one may extend the longevity of their phone's life span. This is done by alleviating older software to be present which would've accumulated junk files. And replacing it with newer more advanced and updated software that is released by Google under the Android Open Source Project[6].

REFERENCES

[1] Chirag Kanthed, Yagyapal Yadav, “Building Custom ROM using AOSP and Improving RAM usage in it”, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Volume2, Issue5, 2017.
[2] Charan K.V, S. P Sharmila, A. S Manjunath, "Customizing AOSP for Different Embedded Devices" 2014 International Conference on Computing for Sustainable Global Development, pp. 259-264, 2014.
[3] Abeer Aljarrah and Mohamed Shehab, “Maintaining User Interface Integrity on Android” IEEE 40th Annual Computer Software and Applications Conference, pp. 449-458, 2016.
[4] Kumar Vimal & Aditya Trevedi, “A Memory Management Scheme for Enhancing Performance of Applications on Android”, IEEE Recent Advances in Intelligent Computational Systems (RAICS),pp 162-166, 2015
[5] Saurabh Manjrekar and Ramesh Bhati, “Custom ROM- A Prominent Aspects of Android”, International Journal of Advanced Research in Computer Engineering and Technology, 2012.
[6] Parikshit Rajput, Vinay Koraganti, Biswajeet Champaty, “Custom ROM”, International Journal of Advances in Science Engineering and Technology, ISSN(p): 2321 –8991, ISSN(e): 2321 –9009 Volume-6, Issue-4, Oct.-2018.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY