

Multi Type Operating System for Android Phones

¹Ajit Singh Raghav, ²Ritika Chugh Malik

^{1,2}Dept. of CSE, Sharda University, Greater Noida, UP, India

Abstract

In this paper we have designed and implemented an Android application that will enable users to interact with the windows operating system. This application is in the form of a desktop widget and supports OS version Android 2.3.3 and onwards. Multi OS windows manager will be setup to manage the appropriate phone system. Windows are automatically sized to fill the whole screen. The main focus was to develop and introduce a new application that boots the windows operating system inside the Android environment. It runs like original windows and provides all features of windows operating system.

Keywords

OHA, Software Development Kit, Application Programming Interface, Dalvik Debug Monitor Server, Virtual Machine

I. Introduction

The computer is in constant evolution since the middle of the 20th century. Computers are developed to perform more advanced calculations, requiring less input power and being more portable [1]. In last few years mankind has witnessed a revolution in the smart phone industry and emerging growth in the usage of mobile applications that range from entertainment and educational apps to health care apps, simple games and more.

Today's Android platform is the most widely used operating system on smart phones and tablet devices are important target for mobile application developers and hardware manufacturers. The aim of this paper was to introduce an android platform and develop an android application that works with windows operating system. The application will be used to provide features of windows operating system like mailing, accessing office documents and more.

II. Development and Task Goals

The main objective of this paper was to design and implement an application for the android platform. For deployment, development and test purpose, Multi OS application Sony Xperia Z was used as a primary test device. The application will act as an original windows operating system and provides all features of windows. The java programming language, eclipse and the android software development kit was used as the environment and development tools. Integrated development with eclipse IDE is selected for the development as it offers direct invoking of tools that are used for developing applications through the eclipse android development tools plug-in [2].

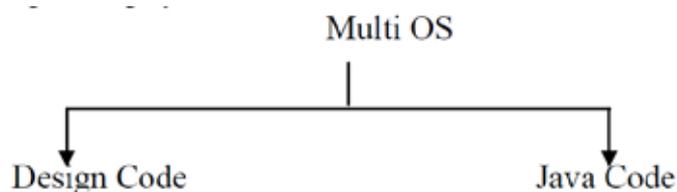
III. Proposed Work

A. Issues in Android Phones

Android based phone do not support all features of windows based phones and we solved it with our application.

B. Objective

Main objective of our application was to run windows operating system inside an android operating system and provide facilities exactly as the real windows operating system.



C. Android Approaches

1. Java Based Approach
2. XML Based Approach
3. Hybrid Approach (Java + XML)

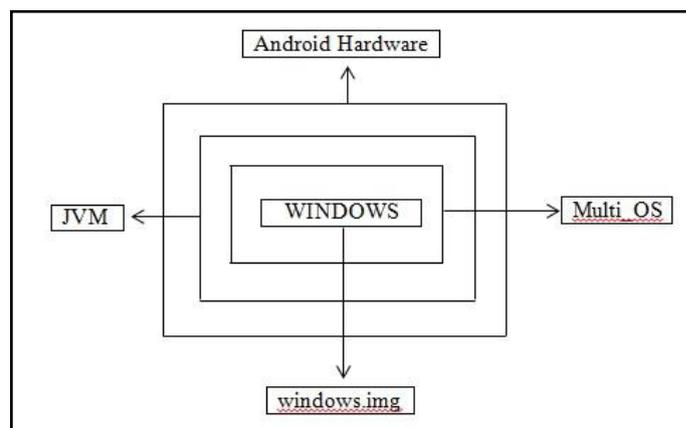


Fig. 1: Application Architecture

D. Two Factors

1. Designing an application

- (i) Layout of the application
- (ii) Controls
- (iii) Text
- (iv) Event Handling :
 - OnClick()
 - OnTouch()
 - OnLongClick()
 - OnKey()
 - OnTouch()

2. Writing Logic for an Application

- We follow the android activity lifecycle for completing work of our application.
- Libsdl.java is main class of application.
- OnCreate () method is starting point of the application and it terminate on OnDestroy ()).

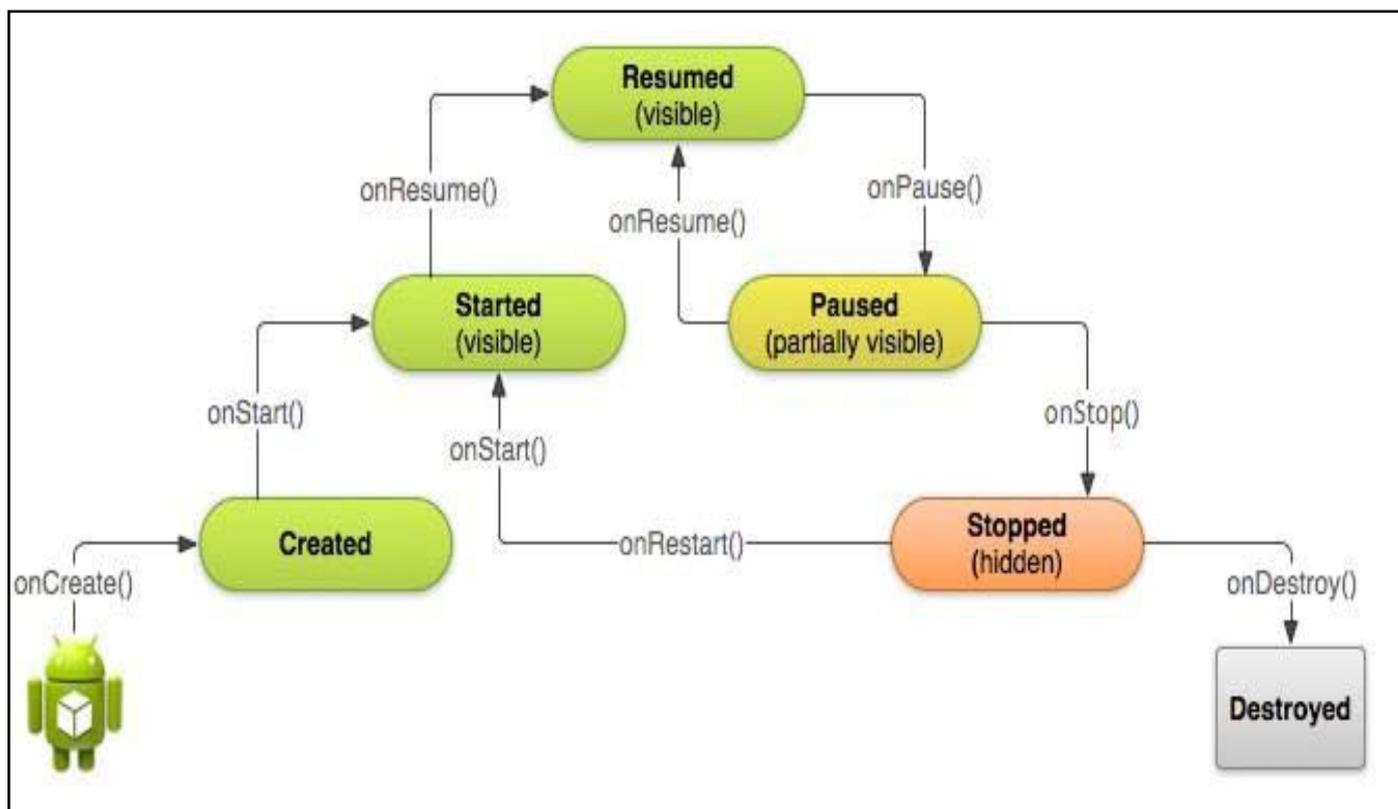


Fig. 2: Activity Lifecycle of Android Application [3]

IV. Designing and Testing

Every developer used XML for designing any application for android platform. It is advanced version of HTML and contains auto generated tags and coding is written in the java programming language. SDK contains a tool called dx which converts java class files into a .dex file means Dalvik Executable file.

The .dex file and resource file (XML files) are packed into an .apk file. The resulting .apk file contains all necessary data to run Android application and can be deployed to an Android device via the adb tool.

Settings and components of application are described in the AndroidManifest.xml file.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest android:versionCode="3" android:versionName="2.5.1" package="Multi.emulator"
  xmlns:android="http://schemas.android.com/apk/res/android">
  <application android:label="@string/app_name" android:icon="@drawable/icon" android:debuggable="true">
    <activity android:label="@string/app_name" android:name="libSDL.multi.libSDL" android:clearTaskOnLaunch="false"
      android:launchMode="singleInstance" android:screenOrientation="sensor"
      android:configChanges="touchscreen|keyboard|keyboardHidden|orientation"
      android:alwaysRetainTaskState="true" android:windowSoftInputMode="stateHidden|adjustPan">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
  <uses-sdk android:minSdkVersion="8" />
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission android:name="android.permission.READ_INPUT_STATE" />
  <uses-permission android:name="android.permission.WAKE_LOCK" />
  <supports-screens android:anyDensity="true" android:smallScreens="true" android:normalScreens="true"
    android:largeScreens="true" />
</manifest>
  
```

Fig. 3: Android Manifest.xml File of Application

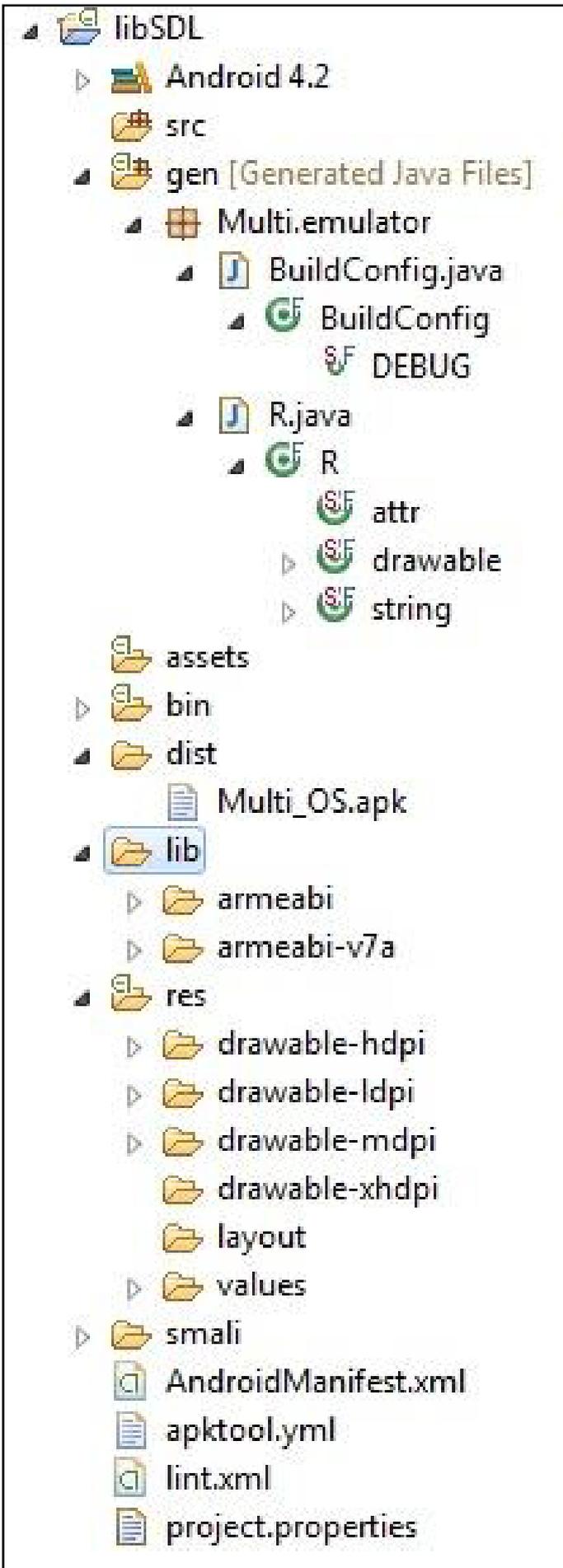


Fig. 4: Folder Structure of Multi OS Application

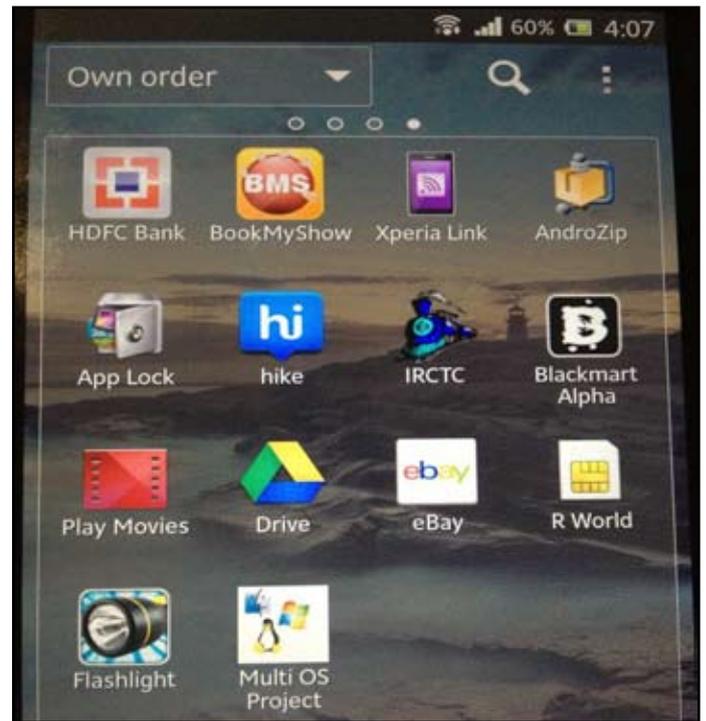


Fig. 5: Phone View of Multi OS After Installation

The objective of this paper was to design and implement an application for the android platform. For deployment, development and test purpose of Multi OS application Sony Xperia Z was used as a test device.

The application will act as an original windows operating system and provides all features of windows.

Fig. below shows the result:



Fig. 6: Windows Boot Screen (1)



Fig. 7: Windows Boot Screen (2)

V. Conclusion and Future Work

The Multi Operating System for Android has been successfully developed. The Operating System can run inside the Android environment. Our application has been successfully tested on Android phone.

During our test runs, we have successfully booted our system. The problem started after booting. Our research was done in building a concept of “The Significance of Multi OS in Android Platform so as to run Windows OS on Android ARM devices”. We know that Windows Operating Systems show better performance on Intel boards. The hardware on which we have tested multi OS is ARM cortex. The ARM does not support Windows Kernel. Our application Multi OS is running successfully but possesses performance issues. To overcome these performance issues the developers of the customized OS for Android must put more effort for kernel performance tuning. We need to over-clock the system. More advanced research must be done so as to improve the heavy calculative tasks. We have to research on an important factor, i.e., “real time” execution of a task in milliseconds. We need to improve the performance of the Multi OS platform. The following tasks could be performed to improve the Performance of Multi OS.

- Interrupts and threads can be allocated to a specific processor via an affinity setting [4].
- Threads are also controllable via scheduling priorities and policies [4].

With the successful addition of the above mentioned factors. We may definitely improve the performance of the running operating system.

Reference

- [1] Benny Skogberg, “Android Application Development- A Guide for the Intermediate Developer”, Degree Thesis, Malmö University, Sweden, Sep. 2010.
- [2] Woubshet Behutiye, “Android ECG Application Development”, Bachelor’s Thesis, BIT, Oulu University of Applied Sciences, 2012.
- [3] Starting an Activity, [Online] Available: <http://www.developer.android.com/training/basics/activity-lifecycle/starting.html>
- [4] Kernel Performance Tuning is Art, not Science (part 1) [Online] Available: <http://www.blog.kreyolys.com/2011/03/05/kernel-performance-tuning-is-art-not-science-part-1/>



Ajit Singh Raghav received his Bachelor Degree in Information Technology from Maharishi Dayanand University, Rohtak, Haryana, India, in 2010 and the Master Degree in Software Engineering from Sharda University Greater Noida, Uttar Pradesh, India, in 2013.



Ritika Chugh Malik worked at an Autonomous Institution (NCCE) as a Lecturer for 2 years. At present, she is working as Assistant Professor in the Department of Computer Science at Sharda University, Greater Noida, Uttar Pradesh, India.