INTRODUCTION

ROMI FADILLAH RAHMAT, B.COMP.SC., M.SC

PEMROGRAMAN MOBILE

LAST UPDATE : 3-SEPTEMBER-2018

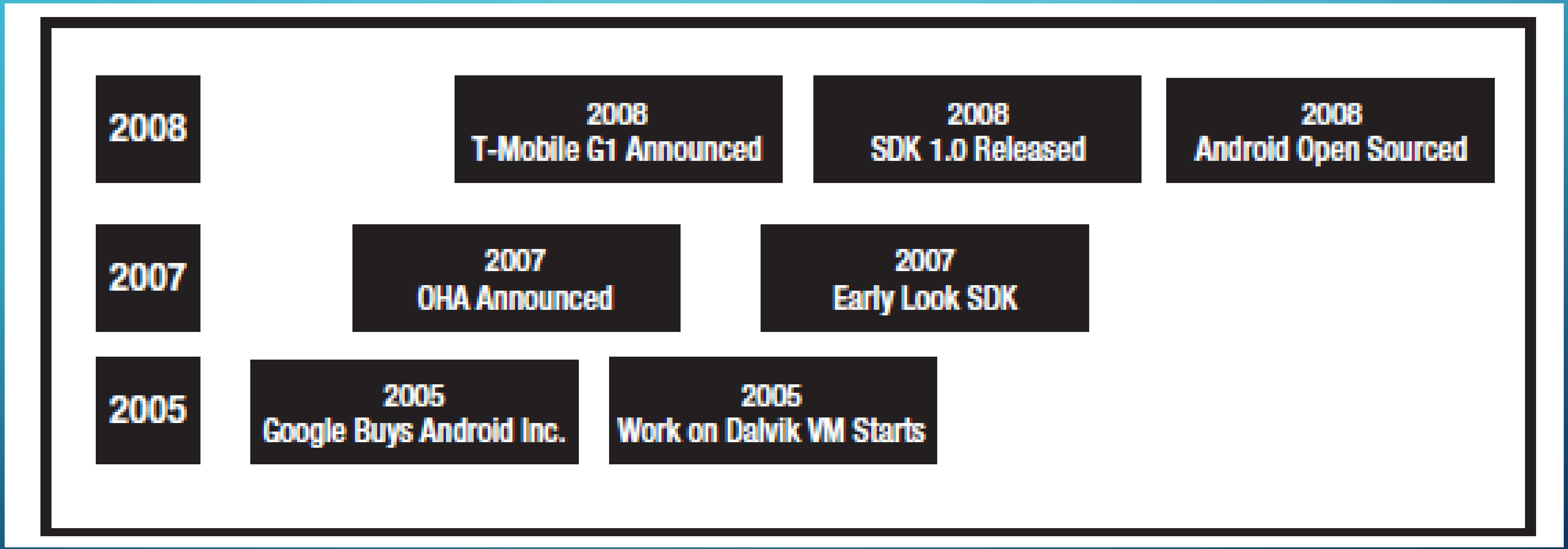
ANDROID ENVIRONMENT

- Android apps had been developed using Eclipse integrated development environment (IDE) with Android Development Tools (ADT) plugin in the past.
- Google introduced Android Studio as the official IDE for Android app development in 2014 and this IDE became the standard.
- Now -> Android Studio 2.3.3

ANDROID HISTORY

- Android is an open-source mobile operating system.
- Architecture of the OS → variant of Linux hence providing extensive security, modularity and productivity at the mobile device level
- Android is developed and maintained by the organization called "Open Headset Alliance" (OHA).
- OHA was established in 2007 with Google being its foremost member.

ANDROID TIMELINE


2008
2008 T-Mobile G1 Announced
2008 SDK 1.0 Released
2008 Android Open Sourced
2007
2007 OHA Announced
2007 Early Look SDK
2005
2005 Google Buys Android Inc.
2005 Work on Dalvik VM Starts

ANDROID TIMELINE

- The Android SDK was first issued as an “early look” release in November 2007.
- In September 2008, T-Mobile announced the availability of T-Mobile G1, the first smartphone based on the Android platform
- In October 2008, Google made the source code of the Android platform available under Apache’s open source license
- When Android was released, one of its key architectural goals was to allow applications to interact with one another and reuse components from one another. Applies to services, data and the user interface (UI).
- As a result, the Android platform has a number of architectural features that keep this openness a reality.

ANDROID TIMELINE

- In late 2008 Google released a handheld device called Android Dev Phone 1 that was capable of running Android applications without being tied to any cell phone provider network.
- In April 2009 release 1.5 SDK, along with a number of other features, such as advanced media-recording capabilities, widgets, and live folders.
- In September 2009 came release 1.6 of the Android OS and, within a month, Android 2.0 followed, facilitating a flood of Android devices

ANDROID HISTORY

- Originally, Android was created by a company called Android Inc. Google acquired this company in 2005.
- After then, Google made it open-source and Android gained a big momentum.
- Android has the market share of around 85% in 2016
- Considering this market share, it is obviously rewarding to invest in Android app development

ANDROID MARKET SHARE

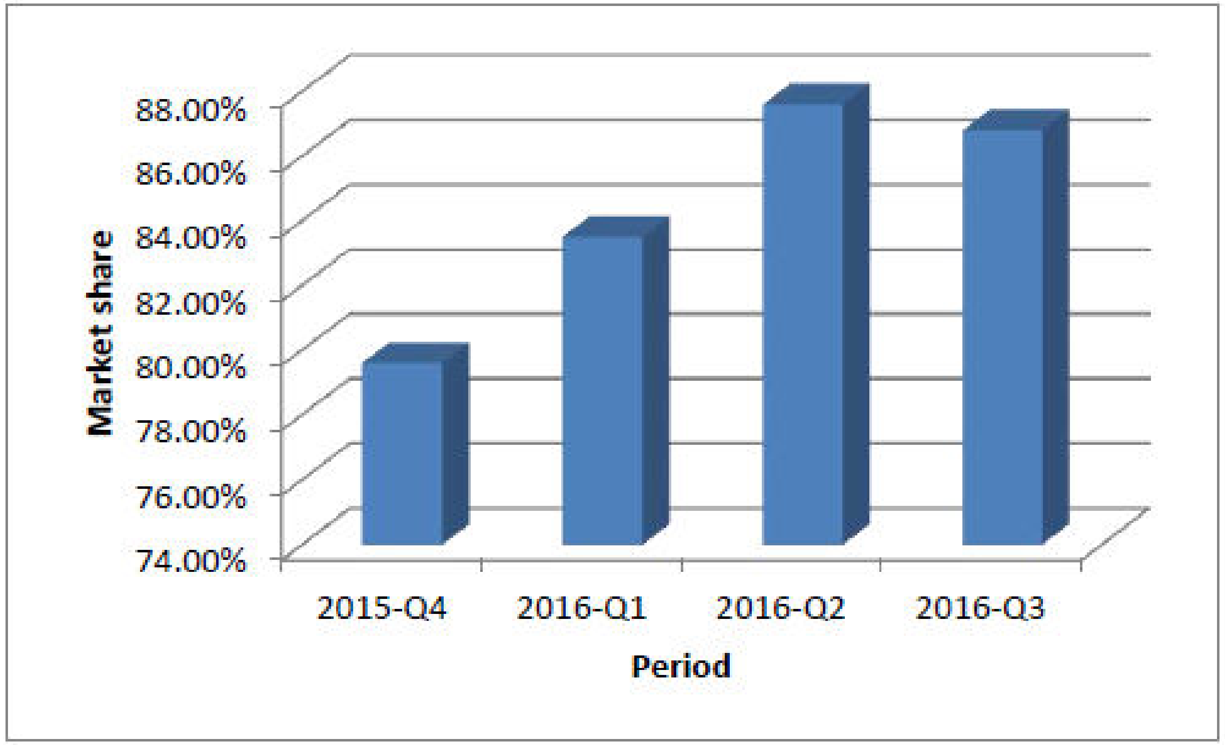


Figure 1.1. Market shares of mobile operating systems between 2015-Q4 and 2016-Q3

ANDROID VERSION

- Android has eight major releases each having several minor revisions. In order to follow these versions easier, developers name them with cookie names.
- The popular versions of Android are Kitkat (Android 4.4), Lollipop (Android 5.1) and Marshmallow (Android 6.0) , Nougat (Android 7.0), and Oreo (Android 8.0).
- Android becomes more capable as the version goes up. However, we have to be careful about selecting the version during app development because not every device uses the latest version. If we develop an app for the Lollipop, it may not run on a device which has Froyo installed.
- Android is utilized not only in smartphones but also in tablets, netbooks, digital television boxes, handheld game devices and even in single board computers.

HOW ANDROID APPS WORK?

- There are different ways the programs run on various platforms. The lowest level software can be written in machine code that runs directly on the microprocessor.
- Since it is difficult to develop complex applications in machine code, operating systems are used. Operating systems provide a communication and control layer between the application software and the hardware

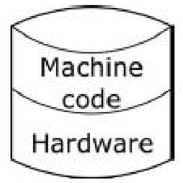


Figure 1.2. Machine code – hardware relation

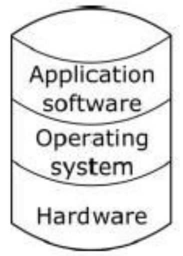


Figure 1.3. Operating system layer between the hardware and the app

COMPILER AND LINKER

- If we want to develop a native application for running on a specific hardware/operating system, we have to do this using a compiler and linker. Compiler and linker takes the source code and creates the executable file that actually runs on the operating system

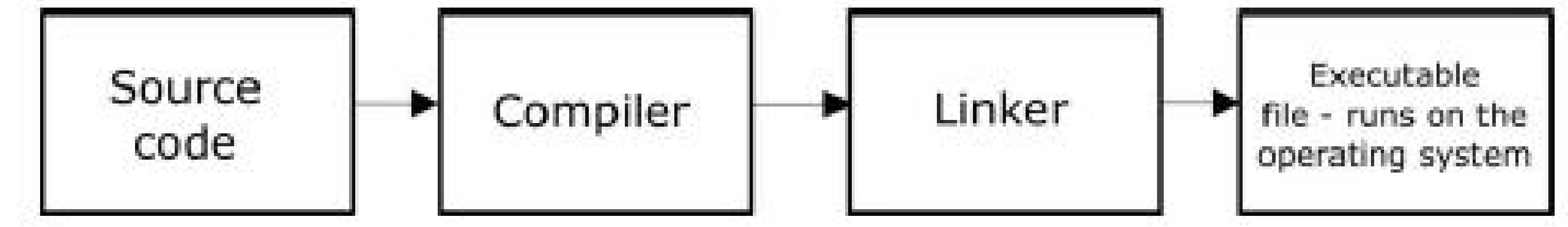


Figure 1.4. Creating a native executable from the source code

- The main advantage of native applications is their speed. However, the disadvantage is the incompatibility across different platforms. For example, we cannot run a native Windows application on Ubuntu and vice versa.
- Virtual machine concept is developed to overcome this limitation. Virtual machine is software that runs on the operating system and provides an abstraction to the developer. The application software runs on top of the virtual machine

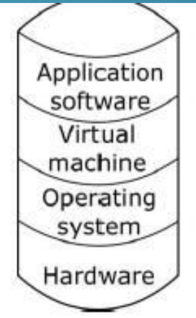


Figure 1.5. Virtual machine between the app and the operating system

JAVA VIRTUAL MACHINE

- Therefore, as long as a computer has the virtual machine running, the application software can run on that computer independent of the hardware and the operating system. A good example is the Java Virtual Machine (JVM). JVM runs on almost all operating systems and platforms. Therefore, when we develop Java software, it will be run on the JVM independent of the operating system/platform.
- The obvious advantage of developing apps that run on virtual machines can then be stated as: “develop once and run on all platforms”. However, applications running on virtual machines are slower than native applications.

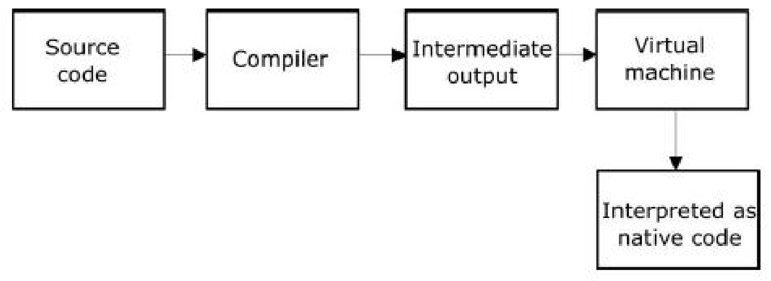


Figure 1.6. Creating an intermediate code from the source code – intermediate code is interpreted by the virtual machine

ANDROID VIRTUAL MACHINE

- Similar to Java applications, Android applications also run on a JVM.
- There are two special virtual machines used in Android: Dalvik Virtual Machine (DVM) and Android RunTime (ART).
- These are specialized JVMs which can run on low system resources.
- The .apk files (executables of Android apps) actually run on these virtual machines.
- DVM has been the default runtime environment (~ virtual machine) until the Lollipop release (Android 5.0).

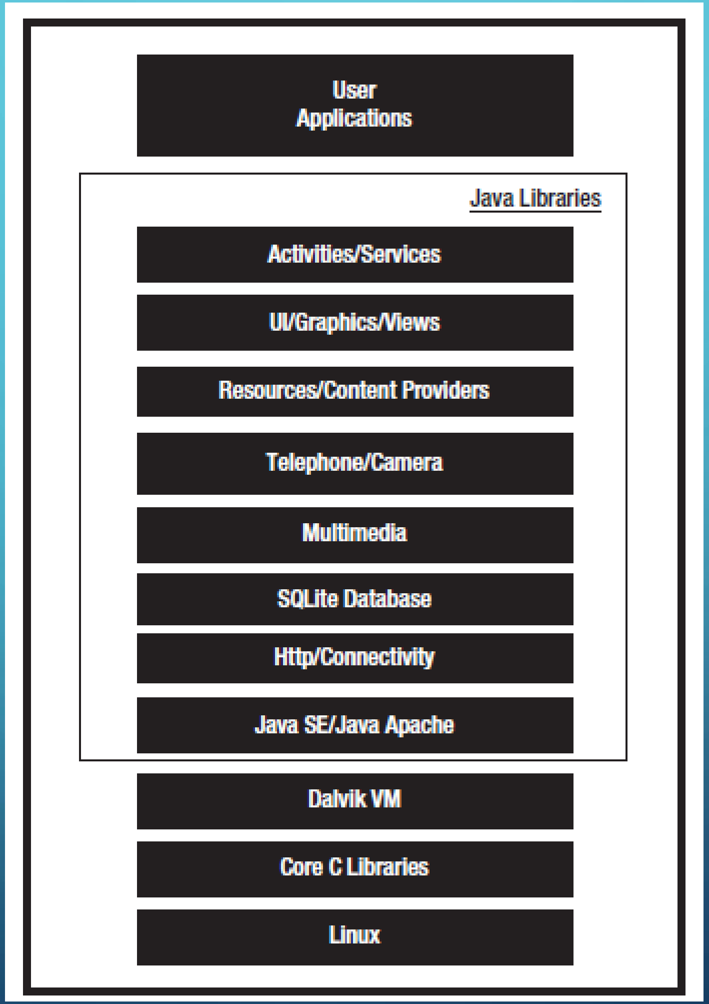
ANDROID VIRTUAL MACHINE

- ART is introduced by Android 4.0 and has been the default VM as of Android 5.0.
- DVM and ART basically do the same job: running Android apps independent of the platform.
- The main advantage of ART over DVM is the utilization of a concept called Ahead of Time (AOT) compilation instead of Just in Time (JIT) approach.
- In AOT, apps are compiled during installation hence they load faster with lower CPU usage. On the other hand, JIT compilation provides lower storage space consumption with relatively longer loading times.

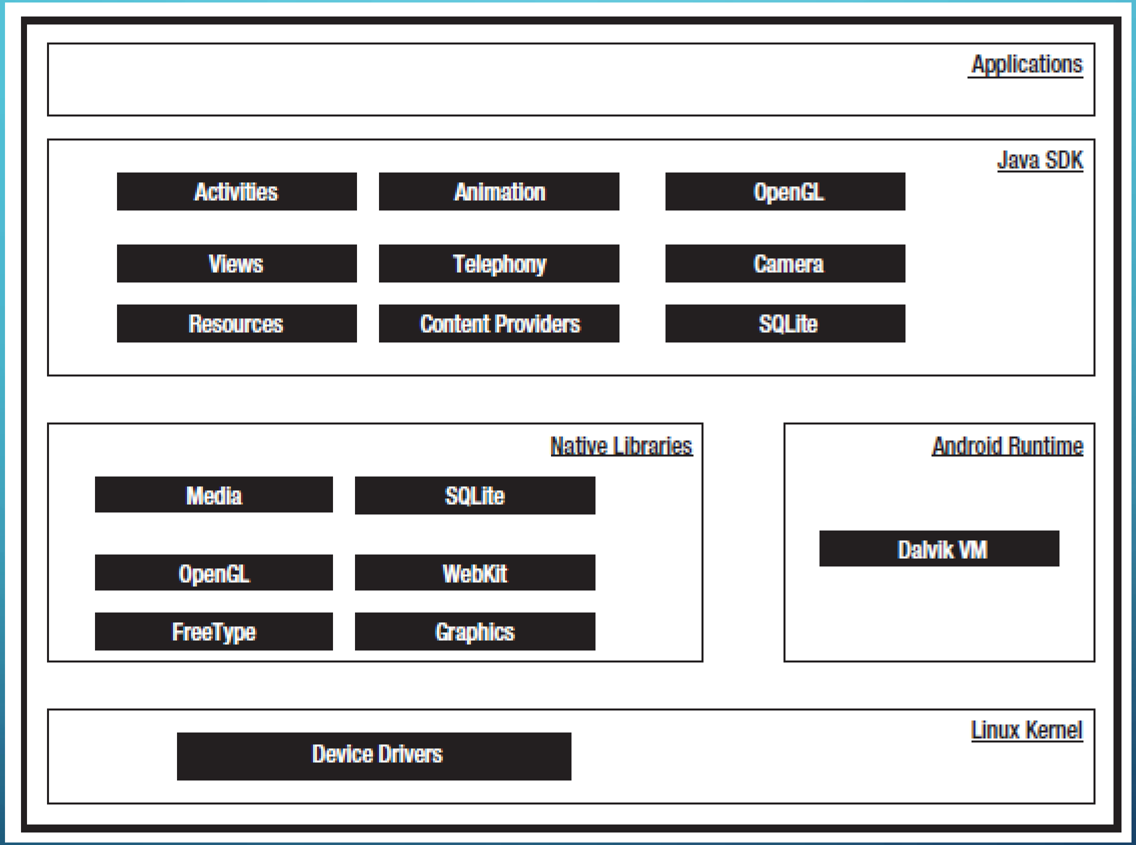
PROGRAMMING LANGUAGE USED IN AND. DEV

- The recommended and convenient way of developing Android apps is using Java programming language. It is used in conjunction with Android Software Development Kit (SDK) in Android Studio environment to develop apps.
- Another official way is using C++ with the Native Development Kit (NDK). This option is used for developing apps with low level instructions such as timing sensitive drivers. With C++ and NDK, we can directly run the app on the Android kernel hence increasing efficiency in exchange of code length and development cost.
- There also exist third-party tools like Xamarin, Crodova and React Native for developing apps. These platforms provide convenience however a native-like performance isn't normally expected from the apps developed by third party tools

HIGH LEVEL VIEW OF ANDROID APPLICATION



ANDROID SOFTWARE STACK



ANDROID SOFTWARE STACK

- At the core of the Android Platform is Linux kernel responsible for device drivers, resource access, power management, and other OS duties. The supplied device drivers include Display, Camera, Keypad, WiFi, Flash Memory, Audio, and IPC (interprocess communication).
- Sitting at the next level, on top of the kernel, are a number of C/C++ libraries such as OpenGL, WebKit, FreeType, Secure Sockets Layer (SSL), the C runtime library (libc), SQLite, and Media
- The system C library based on Berkeley Software Distribution (BSD) is tuned (to roughly half its original size) for embedded Linux-based devices.
- The media libraries are based on PacketVideo's OpenCORE. These libraries are responsible for recording and playback of audio and video formats.
- A library called Surface Manager controls access to the display system and supports 2D and 3D

ANDROID SOFTWARE STACK

- The WebKit library is responsible for browser support; it is the same library that supports Google Chrome and Apple's Safari.
- The FreeType library is responsible for font support
- SQLite is a relational database that is available on the device itself.
- Most of the application framework accesses these core libraries through the Dalvik VM, the gateway to the Android Platform
- Dalvik is optimized to run multiple instances of VMs.
- As Java applications access these core libraries, each application gets its own VM instance

ANDROID SOFTWARE STACK

- The Android Java API's main libraries include telephony, resources, locations, UI, content providers (data), and package managers (installation, security, and so on).
- Programmers develop end-user applications on top of this Java API. Some examples of end-user applications on the device include Home, Contacts, Phone, Browser, and so on.
- Android also supports a custom Google 2D graphics library called Skia, which is written in C and C++.

ANDROID SOFTWARE STACK

- The 3D APIs in Android, are based on an implementation of OpenGL ES from the Khronos group
- From a media perspective, the Android Platform supports the most common formats for audio, video, and images. From a wireless perspective, Android has APIs to support Bluetooth, EDGE, 3G, WiFi, and Global System for Mobile Communication (GSM) telephony, depending on the hardware

ANDROID APP PROJECT

- Proposal and Presentation (5%)
- The proposed apps must be non-trivial, ideally incorporating multiple significant features of the Android platform. The proposed app should satisfy at least one or two of the following categories:
- humanitarian (i.e. it improves quality of life in some significant way)
- monetizable (i.e. it could generate a profit)
- The proposal can include research to similar apps that are available in the Google Play. Students are required to justify that their idea either adds a significant feature, or would be implemented in a way that is a significant improvement over the existing app.
- In the proposal, express the vision and scope of the proposed app.

ANDROID APP PROJECT

- Template of the project proposal:
  - App Name   Names of students in the team
  - Summary of the project (about half page)
  - Explain the major functions of the proposed app and the related theory.
  - Software archtecture and User Interface Design (Some draft screen layouts to get you thinking about layout. The layouts can be handwritten for this milestone.)
  - Market Research with competitive Analysis (who are competitors, how successful are they, how is your idea different)
- All members of the project teams are required to involved in their presentation of the proposed Android Mobile Apps with 10 minutes on week 5.
- One group : 5 persons, consist of boys and girls.