**History of Android Operation System**

Android Inc.

Android Inc. was founded by Andy Rubin, Riche Miner. Early intension was to develop advanced operating system for digital cameras. But the market for the devices was not large enough, so the company diverted its effort towards producing a smartphone operating system.

In July 2005, Google acquired Anroid Inc. for at least $50 million.

**Android** is a [mobile operating system](https://en.wikipedia.org/wiki/Mobile_operating_system) (OS) based on the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) and currently developed by [Google](https://en.wikipedia.org/wiki/Google).  Android is designed primarily for [touchscreen](https://en.wikipedia.org/wiki/Touchscreen) mobile devices such as [smartphones](https://en.wikipedia.org/wiki/Smartphone" \o "Smartphone)and [tablet computers](https://en.wikipedia.org/wiki/Tablet_computer), with specialized user interfaces for televisions ([Android TV](https://en.wikipedia.org/wiki/Android_TV)), cars ([Android Auto](https://en.wikipedia.org/wiki/Android_Auto)), and wrist watches ([Android Wear](https://en.wikipedia.org/wiki/Android_Wear)). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a [virtual keyboard](https://en.wikipedia.org/wiki/Virtual_keyboard).

Despite being primarily designed for touchscreen input, it has also been used in [game consoles](https://en.wikipedia.org/wiki/Video_game_console), [digital cameras](https://en.wikipedia.org/wiki/Digital_camera), regular [PCs](https://en.wikipedia.org/wiki/Personal_computer), and other electronics. As of 2015, Android has the largest [installed base](https://en.wikipedia.org/wiki/Installed_base) of all operating systems.

**What is Kernel?**

A kernel is the lowest level of easily replaceable software that interface with the hardware in our computer. Kernel is responsible for interfacing all of our application that are running in “user mode “ down to the physical hardware and allowing process, known as servers.

**What is SDK?**

It stands for Software Development Kit. It is typically a set of software development tools that allows the creation of application for a certain software packages software framework, hardware platform.

In other words, it is a programming package that enables a programmer to develop applications for a specific platform. SDK includes APIs, programming tools and documentation

API is the way for an application to interact with certain system/application/library etc.

**What is Dalvik and ART?**

Dalvik is a part of software stack that makes up the Android platform. Dalvik is a process virtual machine (like JVM) in Google’s Android operating system that executes applications written for Android.

In general, Dalvik is a virtual machine that runs applications and code written in Java. It is purpose built virtual machine designed specifically for android.

ART is application run time environment used by the Android operating system.

Pros and Cons of Android Operating System

**Pros**

Android is an open source, and it is free for improvement.

It is Google’s, the search engine leader which provides many applications and games free for Android users.

24/7 internet facility. You can have an access to wifi, wherever you go.

You have an option for changing the battery as required.

You can open multipage or multi windows in android.

It is user-friendly and easy to access with numerous tools.

 Endless games and entertainment apps for free and paid.

Android is for pure entertainment and it keep the user alive with updates and many information freely.

 Its USB availability is great compare to its competitors.

 This operating system has a high compatibility that can go with any application.

Unlike iOS of apple, Android is acceptable to many vendors, like HTC, Samsung, Sony etc

Its ROM facility makes it popular among its competitors.

It supports all the cloud services of Google, the largely used services in the internet. It is certainly the greatest advantage.

It easily gets you the information of new SMS, emails, RSS, etc. Notifications are easy to access with the Notification tray on the top

You can download many applications from android market as free and instantly can use them.

Google keep releasing new versions of Android regularly with new features.

Thousands of developers keep developing millions of free and paid apps, so users always have the option to download new apps and never get bored with their device.

Cons

The major complaints come with its battery life which drains out easily.

Without wifi or internet, you can’t access many features of it. You should have a wifi connection at home or at least GPRS through your mobile company.

As it offers many free applications, there are unavoidable advertisements, which make the users tired by making their time wasted.

There are chances of getting malwares or viruses which can affect your device when accessing the free applications like games and entertainments.

In the time of opening multi windows, or installing many applications, the system may go slow and take time.

**Activity Life Cycle**

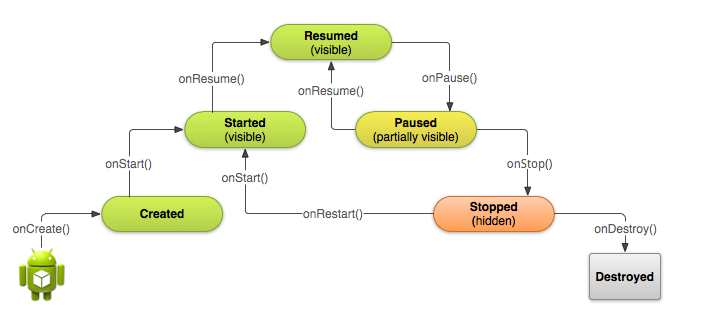


Fig. Android Activity Life Cycle

The major states (methods) in android activity life cycle are

onCreate()

onStart()

onResumer()

onPause()

onStop()

onDestroy()

onCreate()

When we launch an activity in android, it first calls the onCreate() method. onStart() method is called before the activity is being visible to the user. Activity is still not active for user interaction.

onResume()

Activity becomes visible for user to interact with. The activity will be at the top of the activity stack at this point. Now the activity is running/active state and is able to receive user inputs.

onPause()

In the active state, onPause() method will be called when the system is about to resume another activity. On top this one or when the user navigate to some other parts of the system.

Pauses occurs when

The user has pressed the home button.

The device goes to sleep.

Another activity or notification which is on top of it doesn’t obscure the visibility of underlying activity.

When in pause state

The paused activity becomes active by calling onResumed() method.

In case of low memory condition, it gets killed and need to start from beginning calling onCreate() method.

onStop()

This is default action when the user has pressed the back button or a new activity which completely cover it resumes on top. Activity under stopped state is more likely to be killed by the system than one in the pasuse state.

onDestroy()

Release all resource with onDestroy() call.  The system calls [onDestroy()](http://developer.android.com/reference/android/app/Activity.html" \l "onDestroy()) after it has already called [onPause()](http://developer.android.com/reference/android/app/Activity.html" \l "onPause()) and [onStop()](http://developer.android.com/reference/android/app/Activity.html" \l "onStop()) in all situations except one: when you call [finish()](http://developer.android.com/reference/android/app/Activity.html" \l "finish()) from within the [onCreate()](http://developer.android.com/reference/android/app/Activity.html" \l "onCreate(android.os.Bundle)) method

What is intent?

An intent is an abstract description of an operation to be performed. It can be used with [startActivity](http://developer.android.com/reference/android/content/Context.html" \l "startActivity(android.content.Intent)) to launch an [Activity](http://developer.android.com/reference/android/app/Activity.html), [broadcastIntent](http://developer.android.com/reference/android/content/Context.html" \l "sendBroadcast(android.content.Intent)) to send it to any interested [BroadcastReceiver](http://developer.android.com/reference/android/content/BroadcastReceiver.html) components, and [startService(Intent)](http://developer.android.com/reference/android/content/Context.html" \l "startService(android.content.Intent)) or [bindService(Intent, ServiceConnection, int)](http://developer.android.com/reference/android/content/Context.html" \l "bindService(android.content.Intent, android.content.ServiceConnection, int)) to communicate with a background [Service](http://developer.android.com/reference/android/app/Service.html).

An Intent provides a facility for performing late runtime binding between the code in different applications. Its most significant use is in the launching of activities, where it can be thought of as the glue between activities. It is basically a passive data structure holding an abstract description of an action to be performed.