Mobile Operating Systems

An OS is a software interface that is responsible for managing and operating hardware units and assisting the user to use those units. For mobile phones, OSs have been developed to enable users to use phones in much the same way as personal computers were used 1 or 2 decades ago. The most well-known mobile OSs are Android, iOS, Windows phone OS, and Symbian.

#### **Android Operating System**

Android is an open-source mobile OS developed by Google and launched in 2008 [8]. Android is a Linux-based OS that uses Linux 2.6 to provide core services such as security, memory management, process management, network stack, and a driver model. It offers a wide range of libraries that enable the app developers to build different applications. Android applications are usually written in Java programming language

#### **Apple iOS**

Apple iOS is a closed-source code mobile phone OS developed by Apple in 2007; it is used by Apple-only products (iPhone, iPod, and iPad). The iOS architecture is based on three layers incorporated with each other. Cocoa touch is a layer that provides some basic infrastructure used by applications. The second layer is the media layer, which provides audio services, animation video, image formats, and documents in addition to providing two-dimensional (2D) and 3D drawings and audio and video support. The third layer is the core OS, which provides core services such as low-level data types, start-up services, network connection, and access

#### **Symbian Operating System**

Symbian OS is an open-source mobile OS written in C++ programming language developed by Symbian Ltd. in 1977; it is mostly used by Nokia phones. Symbian OS consists of multiple layers such as OS libraries, application engines, MKV, servers, Base-kernel, and hardware interface layer. Symbian was the most prevalent mobile device OS until 2010, when it was taken over by Android

#### **Windows Phone Operating System**

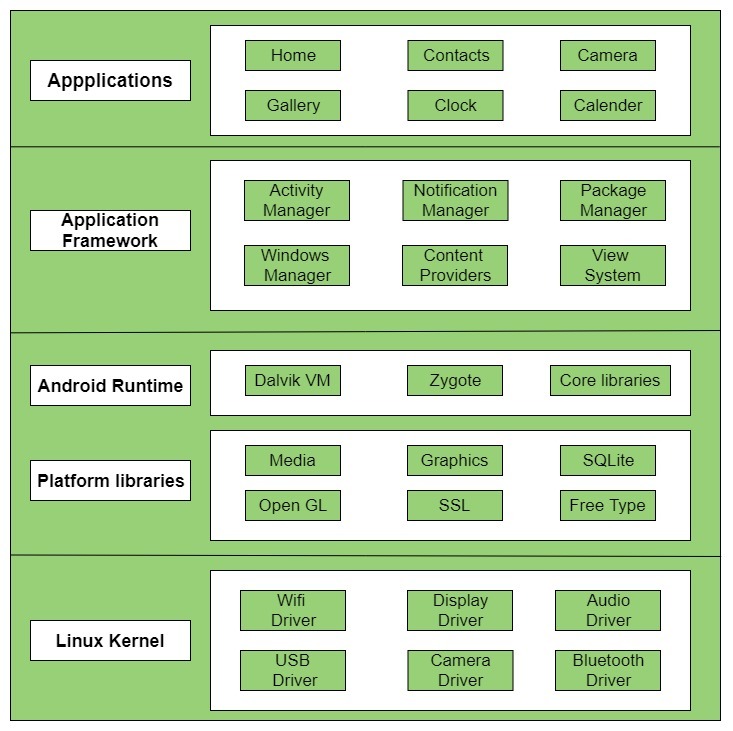
Windows phone OS is a closed-source code mobile OS developed by Microsoft Corporation and used by multiple smart devices (personal digital assistants, smartphones, and touch devices). Windows phone OS is based on a compact version of .Net framework, which gives it an advantage in developing .Net-oriented mobile applications

**Android OS architecture**

Among all the components Linux Kernel provides main functionality of operating system functions to smartphones and Dalvik Virtual Machine (DVM) provide platform for running an android application.

The main components of android architecture are following: -

* Applications
* Application Framework
* Android Runtime
* Platform Libraries
* Linux Kernel



## Applications –

Applications is the top layer of android architecture. The pre-installed applications like home, contacts, camera, gallery etc and third-party applications downloaded from the play store like chat applications, games etc. will be installed on this layer only.  
It runs within the Android run time with the help of the classes and services provided by the application framework.

## Application framework –

Application Framework provides several important classes which are used to create an Android application. It provides a generic abstraction for hardware access and also helps in managing the user interface with application resources. Generally, it provides the services with the help of which we can create a particular class and make that class helpful for the Applications creation.

It includes different types of services activity manager, notification manager, view system, package manager etc. which are helpful for the development of our application according to the prerequisite.

It provides a lot of classes and interfaces for android application development.

This layer and the layer above it i.e. the Application layer are written completely in Java. The applications framework provides all of the major APIs that the applications will use including things like sharing data, accessing the telephony system, and receiving notifications.

## Application runtime –

Android Runtime environment is one of the most important parts of Android. It contains components like core libraries and the Dalvik virtual machine (DVM). Mainly, it provides the base for the application framework and powers our application with the help of the core libraries.

Like Java Virtual Machine (JVM), **Dalvik Virtual Machine (DVM)** is a register-based virtual machine and specially designed and optimized for android to ensure that a device can run multiple instances efficiently. It depends on the layer Linux kernel for threading and low-level memory management. The core libraries enable us to implement android applications using the standard JAVA or Kotlin programming languages.

DVM is like JVM but it is optimized for mobile devices. It consumes less memory and provides fast performance.

## Platform libraries or **Native Libraries** –

The Platform Libraries includes various C/C++ core libraries and Java based libraries such as Media, Graphics, Surface Manager, OpenGL etc. to provide a support for android development.

* **Media** library provides support to play and record an audio and video formats.
* **Surface manager** responsible for managing access to the display subsystem.
* **SGL** and **OpenGL** both cross-language, cross-platform application program interface (API) is used for 2D and 3D computer graphics.
* **SQLite** provides database support and **Free Type** provides font support.
* **Web-Kit** This open-source web browser engine provides all the functionality to display web content and to simplify page loading.
* **SSL (Secure Sockets Layer)** is security technology to establish an encrypted link between a web server and a web browser

## Linux Kernel –

Linux Kernel is heart of the android architecture. It manages all the available drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc. which are required during the runtime.

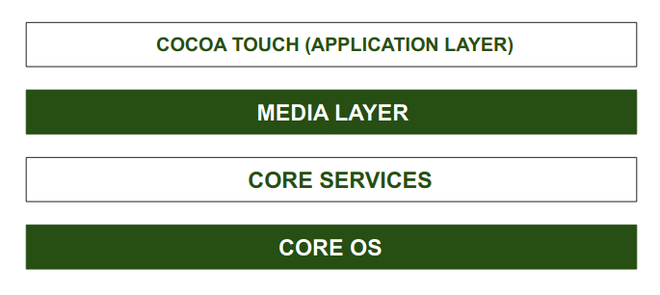
The Linux Kernel will provide an abstraction layer between the device hardware and the other components of android architecture. It is responsible for management of memory, power, devices etc.

The features of Linux kernel are:

* **Security:** The Linux kernel handles the security between the application and the system.
* **Memory Management:** It efficiently handles the memory management thereby providing the freedom to develop our apps.
* **Process Management:** It manages the process well, allocates resources to processes whenever they need them.
* **Network Stack:** It effectively handles the network communication.
* **Driver Model:** It ensures that the application works properly on the device and hardware manufacturers responsible for building their drivers into the Linux build

**Linux kernel** is responsible for device drivers, power management, memory management, device management and resource access.

# Architecture of IOS Operating System



**CORE OS Layer:**  
All the IOS technologies are built under the lowest level layer i.e. Core OS layer. These technologies include:

1. Core Bluetooth Framework
2. External Accessories Framework
3. Accelerate Framework
4. Security Services Framework
5. Local Authorization Framework etc.

It supports 64 bit which enables the application to run faster.

**CORE SERVICES Layer:**  
Some important frameworks are present in the CORE SERVICES Layer which helps the iOS operating system to cure itself ad provide better functionality. It is the 2nd lowest layer in the Architecture as shown above. Below are some important frameworks present in this layer:

1. **Address Book Framework-**  
   The Address Book Framework provides access to the contact details of the user.
2. **Cloud Kit Framework-**  
   This framework provides a medium for moving data between your app and iCloud.
3. **Core Location Framework-**  
   This framework helps to provide the location and heading information to the application.
4. **Core Motion Framework-**  
   All the motion-based data on the device is accessed with the help of the Core Motion Framework.
5. **HealthKit Framework-**  
   This framework handles the health-related information of the user.
6. **HomeKit Framework-**  
   This framework is used for talking with and controlling connected devices with the user’s home.

**MEDIA Layer:**  
With the help of the media layer, we will enable all graphics video, and audio technology of the system. This is the second layer in the architecture. The different frameworks of MEDIA layers are:

1. **ULKit Graphics-**  
   This framework provides support for designing images and animating the view content.
2. **Core Graphics Framework-**  
   This framework support 2D vector and image-based rendering ad it is a native drawing engine for iOS.
3. **Core Animation-**  
   This framework helps in optimizing the animation experience of the apps in iOS.
4. **Media Player Framework-**  
   This framework provides support for playing the playlist and enables the user to use their iTunes library.
5. **AV Kit-**  
   This framework provides various easy-to-use interfaces for video presentation, recording, and playback of audio and video.
6. **Open AL-**  
   This framework is an Industry Standard Technology for providing Audio.
7. **Core Images-**  
   This framework provides advanced support for motionless images.
8. **GL Kit-**  
   This framework manages advanced 2D and 3D rendering by hardware-accelerated interfaces.

**COCOA TOUCH:**  
COCOA Touch is also known as the application layer which acts as an interface for the user to work with the iOS Operating system. It supports touch and motion events and many more features. The COCOA TOUCH layer provides the following frameworks:

1. **EvenKit Framework-**  
   This framework shows a standard system interface using view controllers for viewing and changing events.
2. **GameKit Framework-**  
   This framework provides support for users to share their game-related data online using a Game Center.
3. **MapKit Framework-**  
   This framework gives a scrollable map that one can include in your user interface of the app.
4. **PushKit Framework-**  
   This framework provides registration support.

**Advantages of IOS Operating System**

1. More secure than other operating systems.
2. Excellent UI and fluid responsive
3. Suits best for Business and Professionals
4. Generate Less Heat as compared to Android.

5. Swift is majorly used for iOS application development.

**Advantages of IOS Operating System**

1. Android Devices available with 100+ languages.
2. Android is an Open Source based Source model.
3. Java and Kotlin are majorly used for Android application development.