24CA205 –MOBILE APPLICATION DEVELOPMENT

UNIT 1

MOBILE PLATFORMS AND APPLICATIONS

1.1. Mobile Device Operating Systems

A **mobile operating system** (or **mobile OS**) is an operating system for smart phones, tablets, PDAs, or other mobile devices. While computers such as the typical laptop are mobile, the operating systems usually used on them are not considered mobile ones as they were originally designed for bigger stationary desktop computers that historically did not have or need specific "mobile" features. This distinction is getting blurred in some newer operating systems that are hybrids made for both uses.

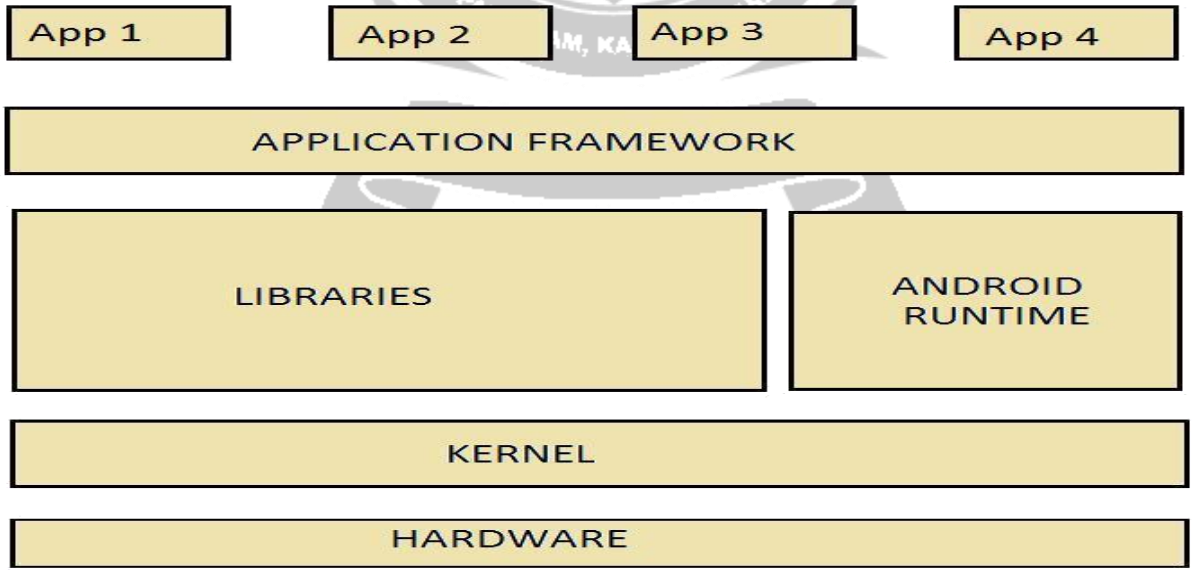
Mobile operating systems combine features of a personal computer operating system with other features useful for mobile or handheld use; usually including, and most of the following considered essential in modern mobile systems; a touch screen, cellular, Bluetooth, Wi-Fi, GPS mobile navigation, camera, video camera, speech recognition, voice recorder, music player, near field communication and infrared blaster.

Mobile devices with mobile communications capabilities (e.g. smartphones) contain two mobile operating systems – the main user-facing software platform is supplemented by a second low-level proprietary real-time operating system which operates the radio and other hardware. Research has shown that these low-level systems may contain a range of security vulnerabilities permitting malicious base stations to gain high levels of control over the mobile device

A mobile operating system, also called a mobile OS, is an operating system that is specifically designed to run on mobile devices such as mobile phones, smartphones, PDAs, tablet computers and other handheld devices. The mobile operating system is the software platform on top of which other programs, called application programs, can run on mobile devices.

Mobile devices with mobile communications capabilities (e.g. smartphones) contain two mobile operating systems – the main user-facing software platform is supplemented by a second low- level proprietary real-time operating system which operates the radio and other hardware. Research has shown that these low-level systems may contain a range of security vulnerabilities permitting malicious base stations to gain high levels of control over the mobile device.

A mobile OS is a software platform on top of which other programs called application programs, can run on mobile devices such as PDA, cellular phones, smartphone and etc. A Mobile operating system is a System Software that is specifically designed to run on handheld devices such as Mobile Phones, PDA's. It is a Platform on top of which the application programs run on mobile devices. Each Operating System follows its own Architecture. Mobile devices evolved the way users across the globe leverage services on the go from voice calls to smart devices which enables users to access value added services anytime and anywhere. At present, the mobile devices are able to provide various services to users but still suffers from issues include Performance, security and Privacy, Reliability and Band width costs. In this paper, we pointed out the issues, challenges, Advantages and Disadvantages of various Mobile Operating systems in terms of their Architectures.



Applications

The diagram shows four basic apps (App 1, App 2, App 3 and App 4), just to give the idea that there can be multiple apps sitting on top of Android. These apps are like any user interface you use on Android; for example, when you use a music player, the GUI on which there are buttons to play, pause, seek, etc is an application. Similarly, is an app for making calls, a camera app, and so on. All these apps are not necessarily from Google. Anyone can develop an app and make it available to everyone through Google Play Store. These apps are developed in Java, and are installed directly, without the need to integrate with Android OS.

Application Framework

Scratching further below the applications, we reach the application framework, which application developers can leverage in developing Android applications. The framework offers a huge set of APIs used by developers for various standard purposes, so that they don't have to code every basic task.The framework consists of certain entities; major ones are:

Activity Manager

This manages the activities that govern the application life cycle and has several states. An application may have multiple activities, which have their own life cycles. However, there is one main activity that starts when the application is launched.

- *Notification Manager*

This manager enables the applications to create customized alerts

- *Views*

Views are used to create layouts, including components such as grids, lists, buttons, etc.

- *Resource Managers*

Applications do require external resources, such as graphics, external strings, etc. All these resources are managed by the resource manager, which makes them available in a standardized way.

- *Content Provider*

  Applications also share data. From time to time, one application may need some data from another application. For example, an international calling application will need to access the user's address book. This access to another application's data is enabled by the content providers.

- *Libraries*

  This layer holds the Android native libraries. These libraries are written in C/C++ and offer capabilities similar to the above layer, while sitting on top of the kernel. A few of the major native libraries include

  Surface Manager: Manages the display and compositing window-ing manager. - Media framework: Supports various audio and video formats and codecs including their playback and recording.

  System C Libraries: Standard C library like libc targeted for ARM or embedded devices.

  OpenGL ES Libraries : These are the graphics libraries for rendering 2D and 3D graphics.

  SQLite : A database engine for Android.

- *Kernel*

  The Android OS is derived from Linux Kernel 2.6 and is actually created from Linux source, compiled for mobile devices. The memory management, process management etc. are mostly similar. The kernel acts as a Hardware Abstraction Layer between hardware and the Android software stack.