Operating System

Assignment 1

Question

Research Task: The structure of OSes (latest stable versions) like iOS, Android, Ubuntu, Windows 10/11, MacOS.

Answer:

Although it's a hard task to tell the exact defination of the operating system, we can regard an OS as a system software that manages and controls the hardware and software resources of a computer system, providing a common services layer for all other software applications to run on top of. It provides the path or the gateway for the users to run multiple applications as it is responsible for allocating system resources like CPU time, memory and input/output devices which acts as a interface for the user to interact with the computer. Operating system is a very critical component of a computer system which we can say that acts like an intermediate between hardware, software and the user. Operating system are divided into various types, and their uses may vary for a certain type of system. In this research task we are mainly focusing on the operating systems that we use in our daily basis like :

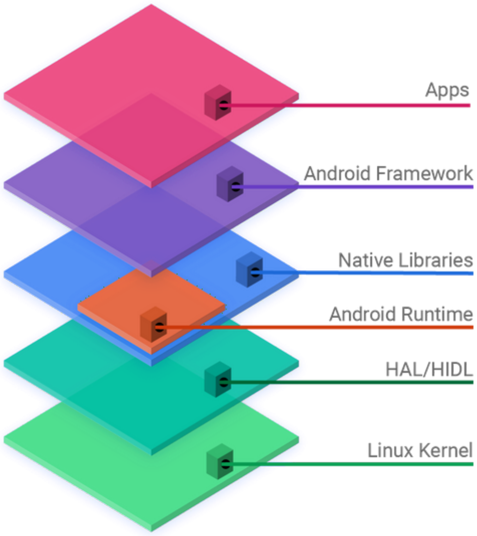
- iOS
- Android
- Ubuntu
- Windows
- MacOS

These are some OSes that we see in our daily lives here and there and some we use it in the daily basis in our own devices. Below we are going to discuss about these OSes structures/architectures in details.

Android Operating System:

Android a commonly used open source operating system that powers a huge amount of modern devices. The original creator of the android operating system is Andy Rubin who founded Android Inc. in Palo Alto, California in October 2003 which is mates Rich Miner, Nick Sears and Chris White which was then later acquired by Google in 2005 for just 50 million at that time which is now a multi billion dollar industry under Google. The source code of android is available to the Android Open Source Project(AOSP) and the first source code was released back in 2007. Since then the development of the android operating of system has been immaculate. Starting from the version 1.0 now we have now reached at the 13th version of the operating system which has been polished every update to make it a perfect operating system. Many companies use the android as the base system which they add their features upon the make it custom but at the core it's the android which powers these phones. Android makes up about $71.74% of the total mobile operating system market according to Google. At the core the programming languages used to make up the android ecosystem are Java, C, C++, Kotlin. On recent developments other languages which the introduction of different frameworks now can be used for cross platform development like React Native for JavaScript, Flutter/Dart and many more.

Structure/Architecture of Android Operating System



> *Figure: Structure of Android Operating System*
>
> *Source: https://en.wikipedia.org/wiki/Android_(operating_system)#/media/File:Android_open_source_project.png*

The architecture of the android operating system can be seen as a stack which consists of various components like the kernal, runtime, libraries, android frameworks and others as seen in the figure above. These components are discussed below:

- Linux Kernel:

  We all know that kernel is the core or the central component of any kind of Operating System. It is the primary interface between the hardware and the processes of the computer. At the core the android kernel is based on an upstream Linux Long Term Supported (LTS) kernel. The LTS kernels are combined with Android-specific patches to form Android Common Kernels (ACKs). The type of kernel that's used in the Android Operating system is the Monolithic Linux Kernel. The main reason to use a Linux Kernel is as Linux is a very portable operating system which can be built and ran in variety of devices. The kernel here used is a modified version of the Linux Kernel, with additional features added to support the mobile devices. Starting from Android 11 the ACKs are used to produce Generic Kernal Images(GKIs). These are 64 bit ARM kernels that can be used in any device with vendor-supplied SoC and driver support which helps the OEMs (Original equipment manufacturers) as they don't have to deal with the kernel and it requires less maintenance.

- HAL/HIDL:

  HAL(Hardware Abstraction Layer) and HIDL(HAL Interface Definition Language) which are one of the components of the architecture are two of the important layers that build up the Android Operating System which provide a standardized interface for communication between the hardware and the software layers of the system. Between the hardware drivers specific to each device and the Android OS framework is the HAL layer. By offering an abstraction layer, it enables the Android framework to interact with the hardware drivers without having to be aware of the precise specifications of the underlying hardware. The HAL layer offers a standardized set of APIs that the Android framework uses to communicate with the hardware, such as the camera, sensors, and input devices. As of Android 10, HIDL is deprecated and Android is migrating to use AIDL everywhere. HIDL is intended to be used for inter-process communication (IPC). HALs created with HDL are called binderized HALs in that they can communicate with other architecture layers using binder inter-process communication (IPC) calls. Usage of HAL/HIDL provides various advantages like portability, abstraction, modularity, compatibility and many others.

- Android Runtime:

  The android runtime is a virtual machine in android. This layer is responsible for the execution of the compiled code of android applications. When the user runs the application then the bytecode written in .dex files are translated by the Android Runtime into the machine code which can be understood by the machine. Upto Android 4.4 Dalvik was used as the virtual machine which was then replaced by ART(android runtime) as the replacement of the Dalvik virtual machine after the release of Android Lollipop. The main difference between these two types of VM is that Dalvik used JIT(Just in time) compilation where as the ART used AOT(ahead of time) compilation. JIT means that the code is translated when the application is ran where as the AOT compilation compiles the code as the software/application is being installed. The AOT approach hugely improved the runtime performance. However there were some drawbacks like apps taking time to get installed, slower system updates and high memory usage. But in the long run as the devices got more powerful this approach made the overall experience better than its predecessor. Other benefits like profile-guided optimization(introduced in android nougat) which helped the optimization as the app ran in common usage patterns. Also AOT provides improved garbage collection which reduces the amount of memory used and boosts the overall performance of the device.

- Native Libraries:

  Android applications can contain compiled native libraries. A native library is code that a developer writes and then compiles for a particular computer architecture. Most of the time that means code written in C or C++. They provide additional functionality or access to system resources not available through the Java-based Android application framework. The Java Native Interface (JNI) allows developers to declare Java methods implemented in native code (usually compiled C/C++). The JNI interface is not specific to the Android operating system, but applies more generally to Java applications running on different platforms. Together, JNI and NDK allow Android developers to implement some features in their applications in native code. Native libraries provide several benefits like:

  - Access to system resources
  - Performance
  - Reusability

  Some popular Native Libraries used in android operating system are OpenCV, webkit, SQLite etc.

- Frameworks:

  The android framework is the set of API's that allow developers to quickly and easily write apps for android phones. It consists of tools for designing UIs like buttons, text fields, image panes, and system tools like intents (for starting other apps/activities or opening files), phone controls, media players, ect. Essentially an android app consists of Activities (programs that the user interacts with), services (programs that run in the background or provide some function to other apps), and broadcast receivers (programs that catch information important to your app). Portions of the framework are publicly accessible through the use of the Android API. Other portions of the framework are available only to OEMs through the use of the system APIs. Android framework code runs inside an app's process. By providing pre-built components and APIs for common tasks, frameworks make it easier for developers to create Android applications and help to ensure that applications are consistent and behave in a predictable manner.