Understanding the Operating System Structure of Android

Introduction:
Android, developed by Google, has emerged as the most widely used mobile operating system globally. Its architecture is structured in a layered manner, with each layer serving specific functions and facilitating the smooth operation of the system. This report aims to provide a detailed exploration of the OS structure of Android, shedding light on its components and their interactions.

1. Kernel Layer:
At the core of the Android OS lies the Linux kernel. The kernel provides essential functionalities such as memory management, process management, device drivers, and security mechanisms. It interacts directly with the hardware, enabling communication with various hardware components including the CPU, memory, input/output devices, and network interfaces. The Linux kernel forms the foundation upon which the entire Android system is built, ensuring stability, security, and hardware abstraction.

2. Libraries Layer:
Built on top of the Linux kernel, the libraries layer comprises a collection of software libraries that provide core functionalities to Android applications. These libraries include the Android Runtime (ART), which executes and manages application code, as well as libraries for multimedia processing, graphics rendering, data storage (such as SQLite), web browsing (WebKit), and secure communication (SSL). These libraries abstract complex tasks, allowing developers to access device features and build robust applications efficiently.

3. Android Runtime (ART):
The Android Runtime (ART) is responsible for executing and managing application code on Android devices. It replaced the Dalvik Virtual Machine in Android 5.0 (Lollipop) and introduced significant performance improvements. ART employs ahead-of-time (AOT) compilation, where application code is compiled into native machine code during installation, resulting in faster application startup times and improved runtime performance. Additionally, ART features enhanced garbage collection mechanisms, leading to better memory management and overall system efficiency.

4. Application Framework Layer:
Sitting atop the libraries layer, the Application Framework provides a set of high-level services and APIs (Application Programming Interfaces) to Android applications. These services include the Activity Manager, responsible for managing application lifecycle; Content Providers, facilitating data sharing between applications; View System, defining

user interface components; Notification Manager, handling notifications; and Package Manager, managing application packages. The Application Framework enables developers to build rich and interactive applications by providing access to system-level functionalities.

5. Application Layer:
The Application Layer represents the topmost layer of the Android OS, where user-installed applications reside. This layer hosts a diverse range of applications, including productivity tools, social media apps, games, utilities, and more. Additionally, it includes pre-installed system applications such as the Phone app, Contacts app, Messaging app, and Browser. Applications in this layer leverage the services and features provided by the underlying layers to deliver a seamless user experience.

6. Security Features:
Security is a fundamental aspect of the Android OS architecture. Multiple layers of security mechanisms are implemented to safeguard user data and ensure system integrity. These include the Linux-based security model, which enforces user and process permissions; application sandboxing, which isolates each application's data and code execution environment; the permissions model, which grants or denies access to sensitive device resources; verified boot, which ensures the integrity of the boot process; and SELinux (Security-Enhanced Linux), which provides mandatory access controls and enhances overall system security.

7. Device-specific Layers:
The Android OS is designed to run on a wide range of devices with varying hardware configurations. As such, device-specific layers are incorporated into the system architecture to accommodate OEM (Original Equipment Manufacturer) customizations and hardware variations. These layers include hardware abstraction layers (HALs) that provide standardized interfaces for accessing device hardware, as well as drivers for specific hardware components such as the camera, sensors, audio, and connectivity modules (Wi-Fi, Bluetooth, etc.). Device-specific layers ensure compatibility and enable seamless integration of hardware components with the Android platform.

In conclusion, the Operating System structure of Android is characterized by its layered architecture, with each layer serving a distinct purpose and contributing to the overall functionality of the system. From the kernel layer providing low-level hardware interactions to the application layer hosting user-installed applications, Android's architecture is designed to be robust, flexible, and scalable. By understanding the intricacies of its structure, developers and users alike can appreciate the complexity and sophistication of the Android operating system.

ASSIGNMENT-01

Operating systems

Name: Nazeeb Ahmed Chowdhury
Roll: BSSE-1432
Date:1/02/2024

Submit to:
Professor Dr. Naushin Nower,
IIT, University of Dhaka