Index

Operating System:

An operating system, or OS, is a software program that manages computer hardware and software resources and provides common services for computer programs. It acts as a bridge between computer hardware and software applications, making it possible for applications to interact with the hardware components of a computer.

Windows OS:-

Windows is a family of operating systems developed by Microsoft Corporation. It is the most widely used operating system in the world for personal computers, ranging from desktops, laptops, tablets, to servers. Windows is known for its graphical user interface (GUI) and user-friendly design, making it easy for users to navigate and perform various tasks on their computer.

Linux OS:-

Linux is a free and open-source operating system that is based on the Unix operating system. It was created by Linus Torvalds in 1991 and has since become one of the most popular operating systems in the world. Linux is known for its stability, security, and flexibility, and is used in a wide range of applications, from servers and supercomputers to smartphones and embedded systems.

S.NO	Linux	Windows
1.	Linux is a open source operating system.	While windows are the not the open source operating system.
2.	Linux is free of cost.	While it is costly.
3.	It's file name case-sensitive.	While it's file name is case-insensitive.
4.	In linux, monolithic kernel is used.	While in this, micro kernel is used.
5.	Linux is more efficient in comparison of windows.	While windows are less efficient.

Java JDK:

Java JDK, or Java Development Kit, is a software development environment used to develop and deploy Java applications. It includes a set of tools that allow developers to create, compile, and run Java programs.
The JDK includes the Java Runtime Environment (JRE), which is needed to run Java applications, as well as a set of tools for development purposes, such as a compiler, debugger, and other utilities. The JDK also includes the Java API, or Application Programming Interface, which provides a set of pre-written code modules that developers can use in their applications.

The Android SDK:

The Android SDK, or Android Software Development Kit, is a collection of software development tools and libraries used to create applications for the Android operating system. It includes a range of tools and resources that developers can use to build, test, and debug Android applications.

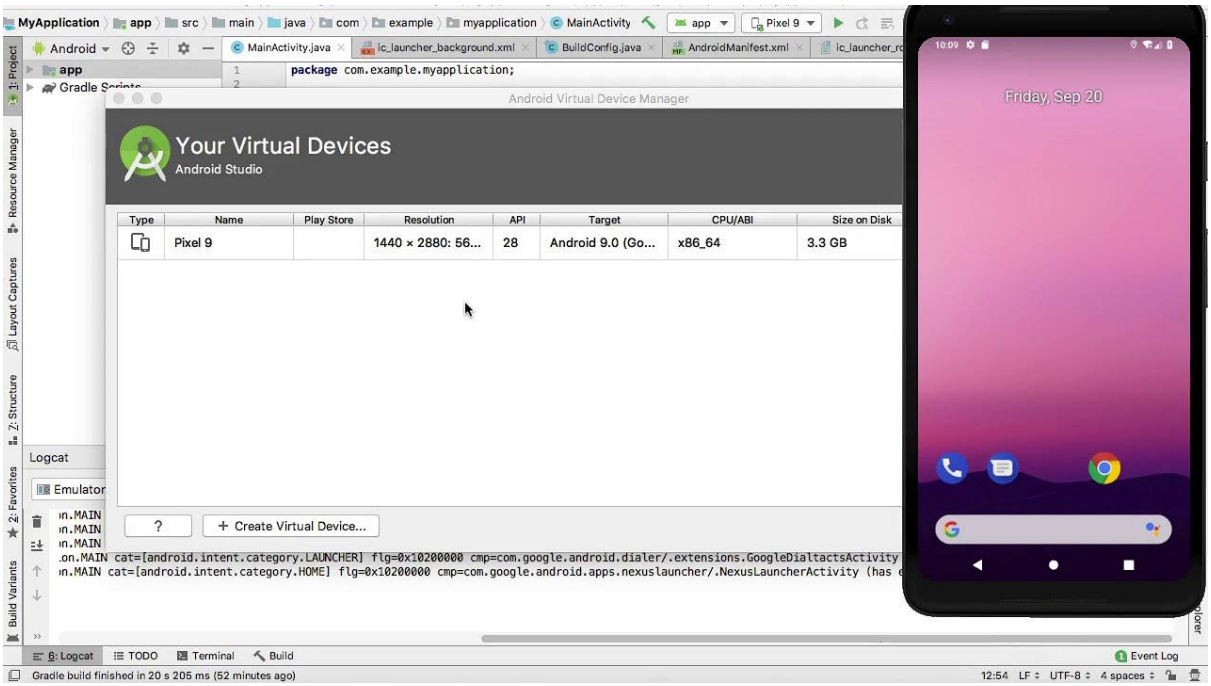
The Android SDK includes several key components, including:
The Android Studio Integrated Development Environment (IDE), which provides a user-friendly interface for developing Android applications.
The Android Debug Bridge (ADB), which allows developers to interact with an Android device or emulator from a computer.
The Android Emulator, which allows developers to test their applications on a virtual Android device before deploying them to a physical device.
The Android SDK Tools, which includes command-line tools for managing the Android SDK, such as the Android Debug Bridge (ADB) and the Android Asset Packaging Tool (AAPT).
The Android SDK Platform-Tools, which includes tools for managing and debugging Android devices and applications.

Android Virtual Device(AVD):

An Android Virtual Device (AVD) is a virtual device configuration used to test and run Android applications on an emulator in the Android Studio IDE. An AVD emulates a specific Android device, allowing developers to test their applications on different screen sizes, resolutions, and hardware configurations.

When creating an AVD, developers can choose the target Android version, screen size, screen resolution, and other device specifications. This allows them to simulate the behavior of their applications on a wide range of devices without needing to have physical devices on hand.

Once an AVD has been created, it can be launched in the emulator, where developers can install and test their applications just as they would on a physical device. The AVD includes an emulated version of the Android operating system and hardware components, such as a CPU, memory, storage, and sensors.

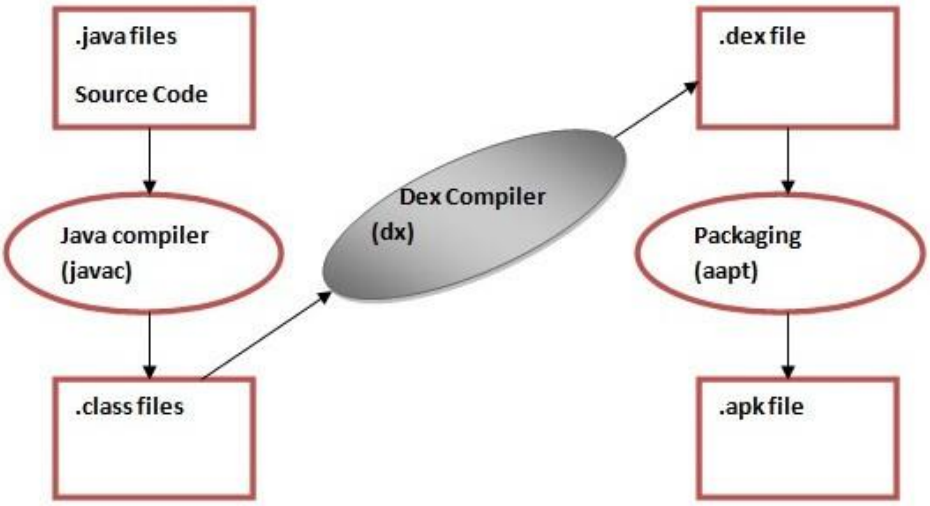


Emulator:

The Android Emulator is a tool included in the Android SDK that allows developers to test and run Android applications on a computer, without the need for a physical Android device. It creates a virtual Android device on the computer that emulates the hardware and software of an actual Android device.

The Android Emulator can be used to test and debug applications across different versions of Android, screen sizes, and hardware configurations. It allows developers to simulate various device features and conditions, such as network latency, device orientation, and battery levels, to identify and resolve issues before deploying their applications on actual devices.

Dalvik Virtual Machine(DVM):
The Dalvik Virtual Machine (DVM) is a virtual machine that was used in the Android operating system to run applications written in the Java programming language. It was developed by Google and named after a fishing village in Iceland, "Dalvík".
The DVM was specifically designed for mobile devices with limited resources, such as memory and processing power. It uses a different bytecode format than the standard Java Virtual Machine (JVM), called Dalvik Executable (DEX), which is optimized for mobile devices.
When an Android application is compiled, its Java code is converted into DEX bytecode, which can then be executed by the DVM. The DVM is responsible for managing memory, running the application's bytecode, and providing access to the device's hardware and software resources, such as sensors and network interfaces.
In later versions of the Android operating system, the DVM was replaced by the Android Runtime (ART), which uses a different approach to executing applications. However, the DEX bytecode format is still used in Android applications, and many of the concepts and optimizations introduced by the DVM are still relevant to Android development.



Difference Between DVM and JVM:

DVM (Dalvik Virtual Machine)	JVM (Java Virtual Machine)
It is Register based which is designed to run on low memory.	It is Stack based.
DVM uses its own byte code and runs the ".Dex" file. From Android 2.2 SDK Dalvik has got a Just in Time compiler	JVM uses java byte code and runs ".class" file having JIT (Just In Time).
DVM has been designed so that a device can run multiple instances of the VM efficiently. Applications are given their own instance.	A single instance of JVM is shared with multiple applications.
DVM supports the Android operating system only.	JVM supports multiple operating systems.
For DVM very few Re-tools are available	For JVM many Re-tools are available.
There is a constant pool for every application.	It has a constant pool for every class.
Here the executable is APK.	Here the executable is JAR.

Installation:
https://www.geeksforgeeks.org/guide-to-install-and-set-up-android-studio/

Components of android application:

An Android application is composed of several components that work together to provide a rich and interactive user experience. The main components of an Android application are:

Activities: Activities are the building blocks of an Android application's user interface. Each activity represents a single screen with a user interface that the user can interact with. Activities are typically used to present content to the user, respond to user input, and manage the application's lifecycle.
Services: Services are background components that run in the background and perform long-running tasks, such as playing music or downloading data. Services can run independently of the user interface and can communicate with other components, such as activities and broadcast receivers.
Broadcast receivers: Broadcast receivers are components that receive and respond to system-wide broadcast messages. Broadcast messages are sent by the system or other applications, and can trigger actions or notifications within an application.
Content providers: Content providers are components that manage and share application data with other applications. They provide a uniform interface for accessing and manipulating data, such as contacts, media files, and settings.

In addition to these core components, Android applications can also use other components, such as fragments, loaders, and intents, to provide additional functionality and features. Each of these components plays a unique role in the overall structure and behavior of an Android application.