

# ANDROID REALLY A LINUX?

Presented by - Akshay Krishna

# Overview

- What is Android?
- Android and Google
- Design Goals
- Android Architecture
- Linux Extensions
- Android Applications
- Summary

# What is Android?

- Informally - We are not talking about robots, but the stuff used in smartphones.
- Formally - Android is relatively a new operating system designed and developed by Google to run on mobile devices.
- Spoiler Alert! - Android is based on the Linux kernel. It introduces only a few new concepts to the Linux kernel, utilizing most of the Linux facilities (processes, user IDs, virtual memory, file systems, scheduling, etc.)
- Android is an open-source platform and is freely available to the manufacturers. It is not only popular with the consumer centric devices (tablets, televisions, etc.), but is increasingly used as the embedded OS for devices that needs GUI (smart watches, automotive dashboards, etc.)

# Android and Google

- Android combines open-source code with closed-source third party applications. The open-source part of Android is called the Android Open Source Project (AOSP).
- The main goal of Android is to support a rich third-party application environment. But as Android is open-source, the device manufacturers can modify the platform (however they want) and compatibility issues arises.
- To solve these compatibility issues, the CDD (Compatibility Definition Document) has been defined.
- CDD describes the ways Android must behave to be compatible with the third party applications. This documents describes what is required to be a compatible Android device.

# Android and Google continued...

- To enforce the compatibility, Android allows additional proprietary services to be created on top of the open-source platform, providing services which are not provided by the platform.
- Google Play is Google's online store for Android apps. Developers publish their applications via Google Play. This enforces responsibility for the proprietary service to ensure the applications compatibility with the device.
- Google Play uses two main mechanisms to ensure compatibility.
  - The first, any device shipping with Android must be a compatible Android device as per the CDD.
  - Second, Google Play must know about any features of a device that an application requires so the application is not made available on devices that lack those features.

# Design Goals

A number of key design goals for the Android platform evolved during its development:

- Provide a complete open-source platform for mobile devices.
- Strongly support proprietary third-party applications with a robust and stable API.
- Allow all third-party applications, including those from Google, to compete on a level playing field (unlike the other company we know of).
- Provide an application security model in which users do not have to deeply trust third-party applications.

# Design Goals continued...

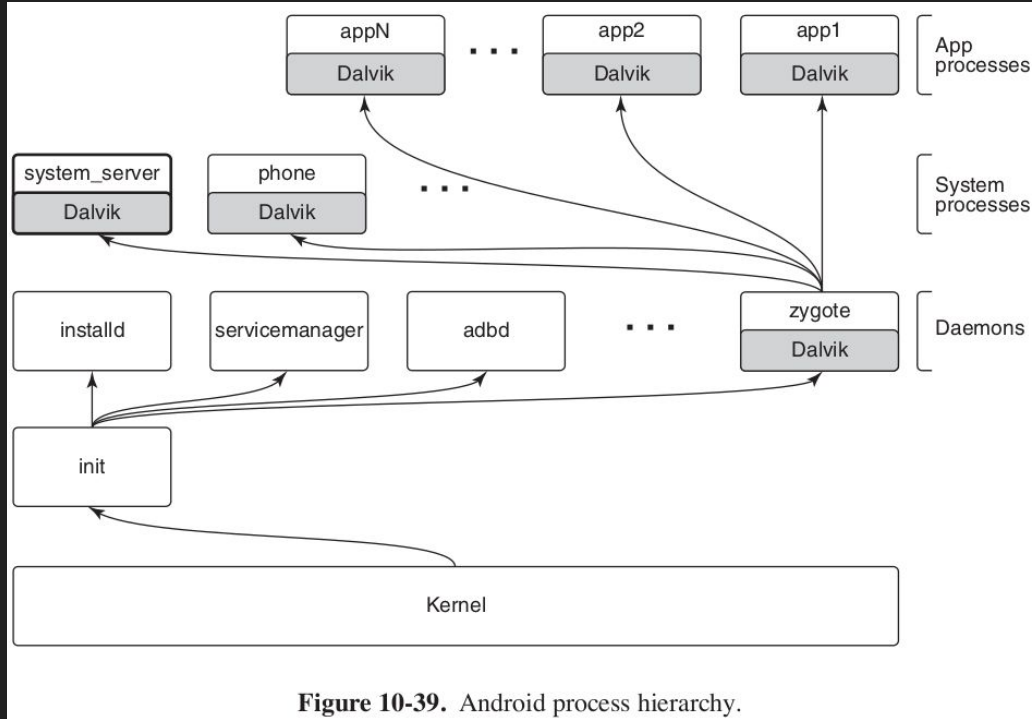
- Support typical mobile user interaction: The system needs to optimize for the cases with fast app launch and switch times.
- Manage application processes for users, simplifying the user experience around applications so that users do not have to worry about closing applications when done with them.
- Encourage applications to interoperate and collaborate in rich and secure way.
- Create a general-purpose operating system. Mobile devices are a new expression of general purpose computing.

# Android Architecture

- Android is built on top of the Linux kernel, with only a few significant extensions to the kernel.
- However, the Android's user-space implementation is quite different from the traditional Linux distribution, and uses many of the Linux features in different ways.
- As in a traditional Linux system, Android's first user-space process is *init*. The daemons Android's *init* process starts are different, focused more on low-level details.
- Android also has an additional layer of processes, running Dalvik's Java language environment, which are responsible for executing all parts of the system implemented in Java.



# Android Architecture continued...



**Figure 10-39.** Android process hierarchy.

# Linux Extensions

- Wake Locks
  - Power management on mobile devices is different than on traditional computing systems. Hence, Android added a new feature to Linux called Wake Locks (also known as suspend blockers).
  - On traditional computing devices, there are two power states: running and ready for user input, or deeply asleep and unable to continue executing. But for mobile devices, the users have different expectations.
  - Wake locks on Android allow the system to go into a deeper sleep mode, without being tied to an explicit user action like turning the screen off.
  - While the screen is on, the system always holds a wake lock that prevents the device from going to sleep, so it will stay running, as we expect.
  - When the screen is off, however, the system itself does not generally hold a wake lock, so it will stay out of sleep only as long as something else is holding one. When no more wake locks are held, the system goes to sleep, and it can come out of sleep only due to a hardware interrupt.

# Linux Extensions continued...

- Out-of-Memory Killer

- Linux includes an “out-of-memory killer” that attempts to recover when memory is extremely low.
- In modern operating systems with paging and swap, it is rare for applications to see out-of-memory failures.
- Android does not have a swap space, so it is much more common to be in out-of-memory situations. Android uses the standard Linux configuration to over-commit memory.
- Android introduces it's own out-of-memory killer to the kernel. The Android's killer runs more aggressively: whenever the RAM is getting low.
- Instead of trying to guess which process has to be killed, the Android's killer relies very strictly on information provided to it by the user-space. The Android's killer relies on the per-process *oom\_adj* parameter to kill the processes. Processes with a higher *oom\_adj* will always be killed before those with a lower ones.

# Android Applications

- Android provides an application model that is different from the normal command-line environment in Linux shell.
- An application is not an executable file with a main entry point; it is a container of everything that makes up that application.
- An Android application is a file with the *apk* extension, for Android Package. This file is actually a normal *zip* archive, containing everything about the application.
- The important contents of an *apk* are:
  - A manifest describing what the application is, what it does, and how to run it.
  - Resources needed by the application, XML data for layouts and other descriptions, graphical bit-maps, etc.
  - The code itself.
  - Signing information, securely identifying the author.

# Summary

- Android is a platform for allowing apps to run on mobile devices.
- It is based on the Linux kernel, but consists of a large body of software on top of Linux, plus a small number of changes to the Linux kernel.
- Android packages are self contained and have a manifest describing what is in the package.