

ANDROID FRAMEWORK INTERVIEW QUESTION

1. What are the key components of the Android Framework?

The key components of the Android Framework are:

Activity Manager

Package Manager

Content Provider

Resource Manager

View System

Notification Manager

Location Manager

Telephony Manager

2. What is AAPT?

AAPT is short for Android Asset Packaging Tool. This tool provides developers with the ability to deal with zip-compatible archives, which includes creating, extracting as well as viewing its contents.

3. What is adb?

Adb is short for Android Debug Bridge. It allows developers the power to execute remote shell commands. Its basic function is to allow and control communication towards and from the emulator port.

4. What is ANR?

ANR is short for Application Not Responding. This is actually a dialog that appears to the user whenever an application has been unresponsive for a long period of time.

Enumerate the steps in creating a bounded service through AIDL.

1. create the .aidl file, which defines the programming interface
2. implement the interface, which involves extending the inner abstract Stub class as well as implanting its methods.
3. expose the interface, which involves implementing the service to the clients.

5. What is AIDL?

AIDL, or Android Interface Definition Language, handles the interface requirements between a client and a service so both can communicate at the same level through interprocess communication or IPC. This process involves breaking down objects into primitives that Android can understand. This part is required simply because a process cannot access the memory of the other process.

6. What data types are supported by AIDL?

AIDL has support for the following data types:

-string

-charSequence

-List

-Map

-all native Java data types like int, long, char, and Boolean

7. What is AOSP?

Google maintains the AOSP, which is an ASCII text file software package development program. Because of the project's nature, anybody may inspect and submit code and patches to the repository. Google is in charge of the

majority of the effort and it's overall roadmap. The AOSP is updated daily with the most recent robot bug and security updates. Every year at its I/O developer conference, Google releases significant new features for the operating system. The most recent version released is Android 12.

8. What is HAL?

☐ Android HAL [Hardware Abstraction Layer] (in the abstract) bridges the gap between hardware and software.

☐ Android application/framework communicates with the underlying hardware through Java APIs not by system calls.

☐ But Linux has the ability to handle only system calls from applications.

☐ Thus we need a glue layer between the Android framework and the Linux system.

☐ Android HAL allows the Android application/framework to communicate with the hardware-specific device drivers.

☐ For Android applications, HAL provides API's through which the service can place a request to the device.

☐ HAL uses functions provided by the lower layer Linux system to service the request from the android framework.

☐ HAL is a c/c++ layer which is a vendor specific implementation.

9. What is HIDL?

HAL interface definition language or HIDL is an interface description language (IDL) to specify the interface between a HAL and its users. HIDL allows specifying types and method calls, collected into interfaces and packages.

Generate the HAL files

To generate the HAL files you need to use the hidl-gen tool run:

```
# PACKAGE=android.hardware.simple@2.0
# LOC=hardware/interfaces/simple/2.0/default/
# make hidl-gen -j64
```

```
# hidl-gen -o $LOC -Lc++-impl -randroid.hardware:hardware/interfaces -randroid.hidl:system/libhidl/transport $PACKAGE
# hidl-gen -o $LOC -Landroidbp-impl -randroid.hardware:hardware/interfaces -randroid.hidl:system/libhidl/transport $PACKAGE
```

10. What is a Device Driver?

- ☐ Device driver is a piece of software that will drive hardware.
- ☐ Linux has predefined system calls which can be used by the user space application to perform the operation on hardware.
- ☐ The part of code which is capable of implementing system call to hardware specific operation is called device driver.

11. What is the role of device management?

Device management is the act and skill of overseeing the installation, operation, and upkeep of a computer or network. It's a wide term that refers to a multitude of administrative tools and actions that are beneficial in keeping a PC or network working smoothly.

Device management in a nutshell!

The following tasks are handled by device management:

- Obtaining & installing device & component-level drivers, as well as associated software.
- Configuring a device to behave as intended with the associated operating system, business/workflow applications, and/or additional hardware components.
- Putting security processes and protections in place.

12. What is a Loader?

A Loader is a component of the Android framework that is used to asynchronously load data from a content provider or other data source, without blocking the UI thread.

13. What is a LoaderManager?

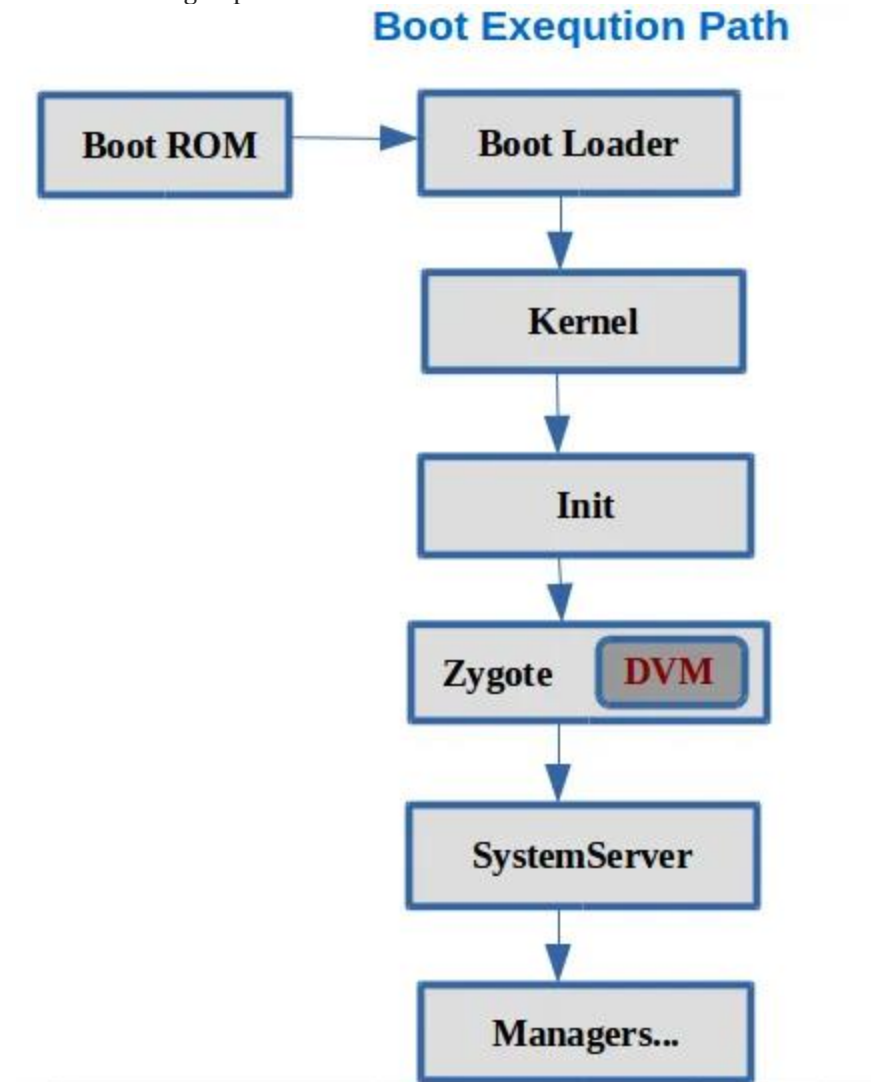
A LoaderManager is a component of the Android framework that manages one or more Loaders, which are used to asynchronously load data from a content provider or other data source.

14. What is an Android SDK?

The Android SDK (Software Development Kit) is a collection of tools and libraries that developers use to create Android applications. It includes tools for building, debugging, and testing Android applications.

15. What is Android Boot Process?

When switching off the power of the Android device and switching on it again, this process is known as the Android Booting sequence.



The above Image shows 5 stages of the Booting process for an Android-powered device:

- 1st Stage is Boot ROM and Boot Loader
- 2nd Stage is Kernel
- 3rd Stage is Init

- 4th Stage is Zygote and DVM
- 5th Stage is SystemServer and Managers

16: WHAT IS HAL?



- 1) Android HAL [**Hardware Abstraction Layer**] (in abstract) bridges the gap between hardware and software.
- 2) Android application/framework communicates with the underlying hardware through Java APIs not by system calls.
- 3) But the Linux has the ability to handle only system calls from the application.
- 4) Thus we need a glue layer between the Android framework and the Linux system.
- 5) Android HAL allows the Android application/framework to communicate with the hardware specific device drivers.
- 6) For Android application, HAL provides APT's through which service can place a request to the device.

7) And HAL uses functions provided by the lower layer Linux system to serve the request from the Android framework.

8) HAL is a c/c++ layer which is a vendor specific implementation.

HAL Architecture

☐ There are two different types of Android HAL architecture:

1) First one is the legacy Android HAL which looks similar to library implementation.

2) Second one gives the complete abstraction and control over the device vendor.

☐ Android HAL implementation looks similar to a Linux device driver. (Since LDD is a HAL for Linux)

☐ Most of the Vendor specific implementations can be done in Android HAL rather than the driver. So that license difference between the driver (Open source license GPL) and the HAL (Apache License) will give more level of abstraction to vendor.

☐ HAL has structures which specify the HAL type, Module type, version detail, and a set of function callbacks (methods) which are registered to the Android framework layer.

☐ For example if a service from Android framework needs data from a sensor device, the service will make a call to corresponding function in sensor HAL which in turn will talk directly with Linux device driver.

☐ The driver will trigger the sensor driver and deliver the data back to the HAL so it can be passed back to Android application.

☐ The HAL module is compiled into a shared object that is dynamically loaded at runtime when needed.