Android build system

Nguyen Tran (Nguyen.TranLeHoang@vn.bosch.com)

Oct-2024

BOSCH

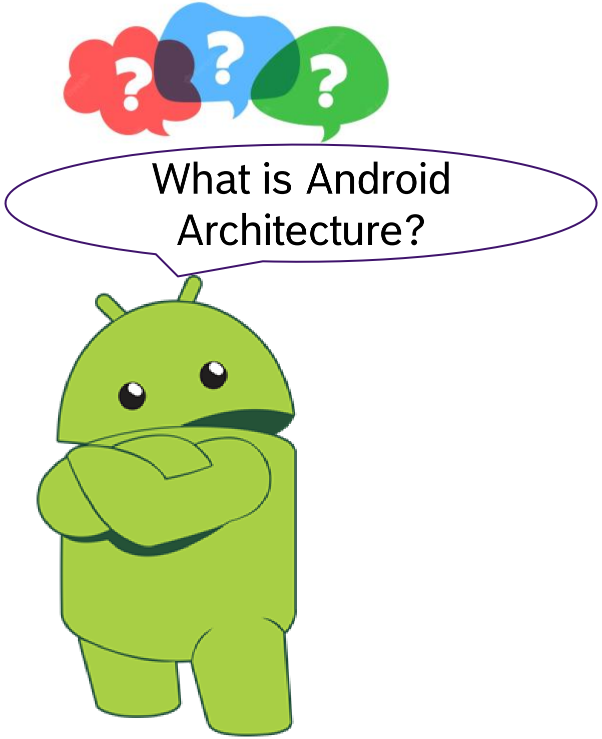
Day 1

QUIZ (1/10)

What is Android?

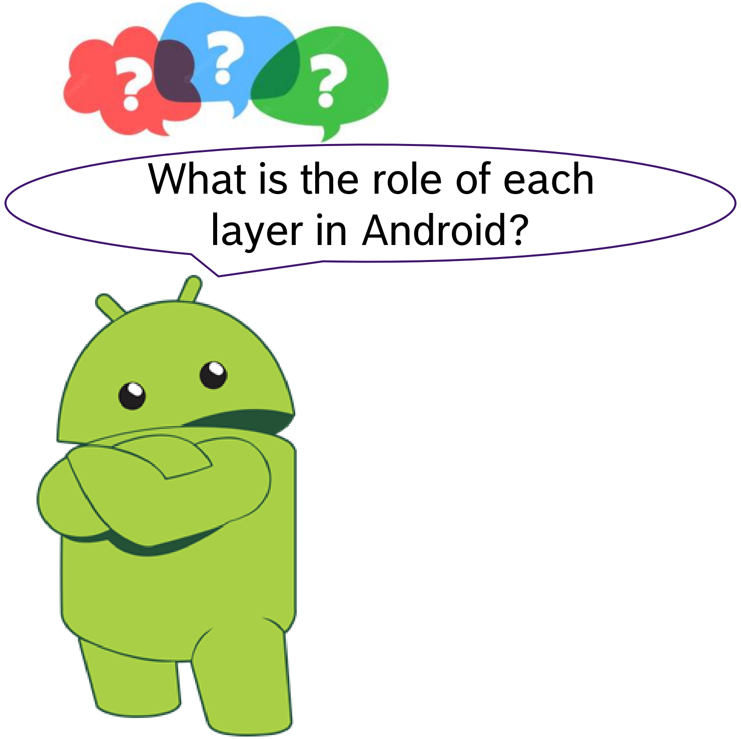
QUIZ (2/10)

QUIZ (3/10)



What is Android Architecture?

QUIZ (4/10)

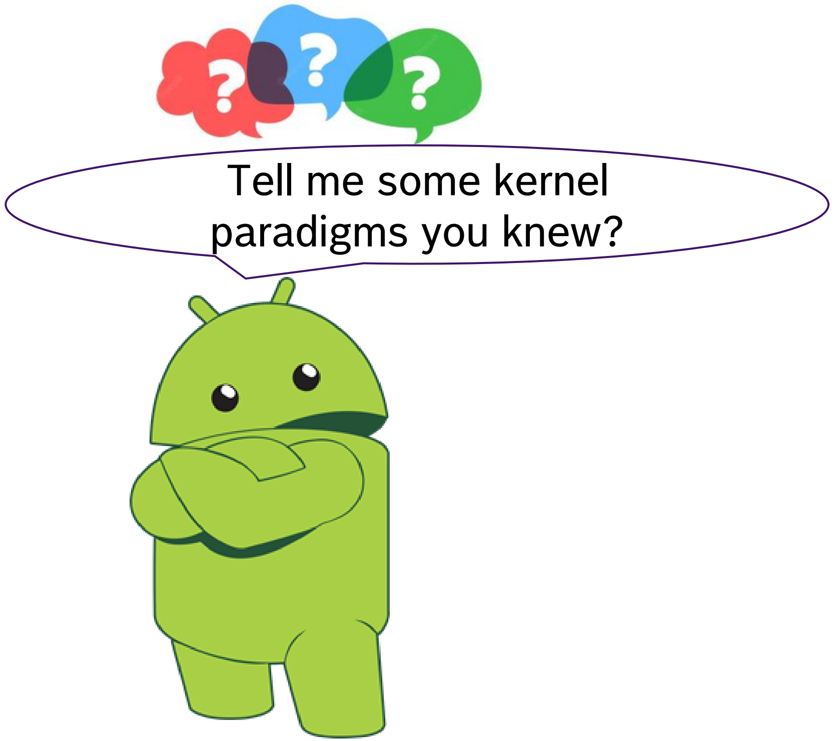


What is the role of each layer in Android?

QUIZ (5/10)

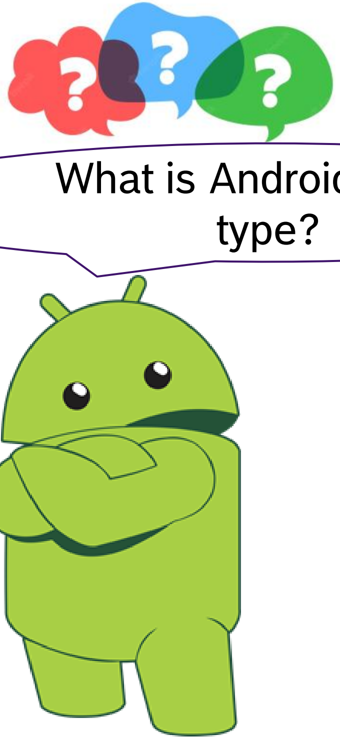
Tell me some Android modification in Linux kernel?

QUIZ (6/10)



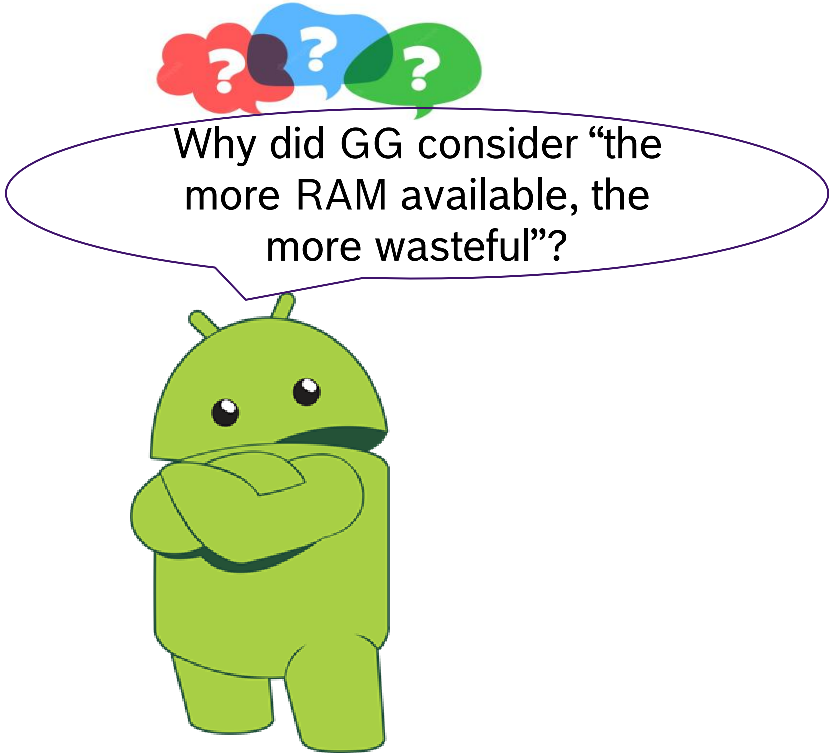
Tell me some kernel paradigms you knew?

QUIZ (7/10)



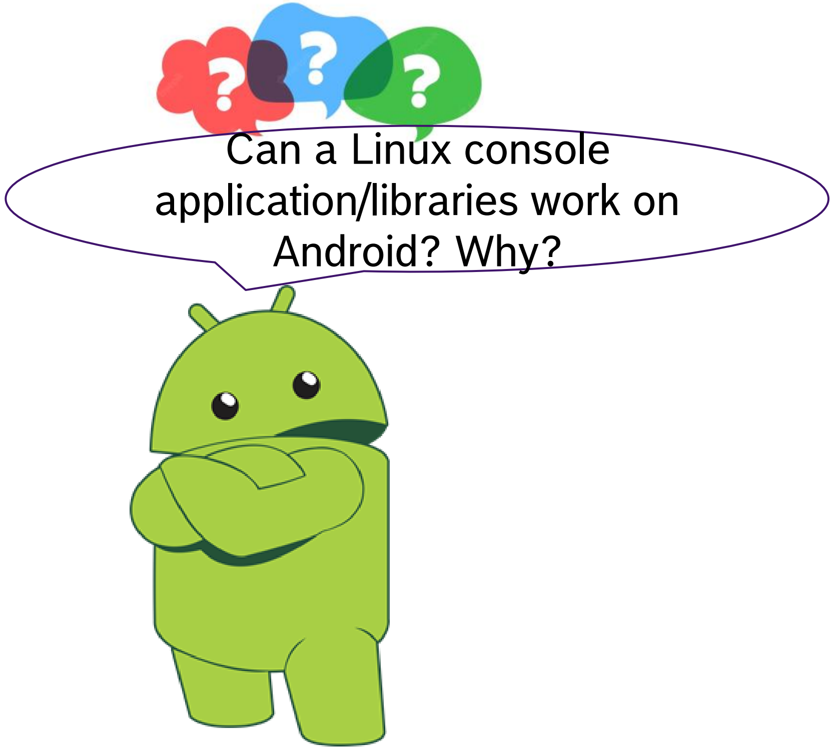
What is Android kernel type?

QUIZ (8/10)



Why did GG consider “the more RAM available, the more wasteful”?

QUIZ (9/10)



Can a Linux console application/libraries work on Android? Why?

QUIZ (10/10)

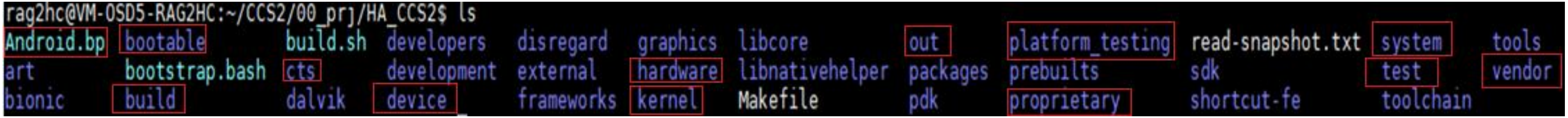
Why shouldn’t we use shared libraries directly for an Android service?

Group working

Let’s discuss: What is Android build system?

Android build system

- The build system uses some pre-set environment variables and a series of **'make'** files in order to build an Android system and prepare it for deployment to a platform.
- **Android.bp** and **Android.mk** are being used now in Android build system.
- There is only one official file named 'Makefile', at the top of the source tree for the whole repository:
  - We can **set some environment variables**, then type **'make'** or just **m** to build stuff.
  - We can add some options to the make command line (other targets) to turn on verbose output or perform different actions.

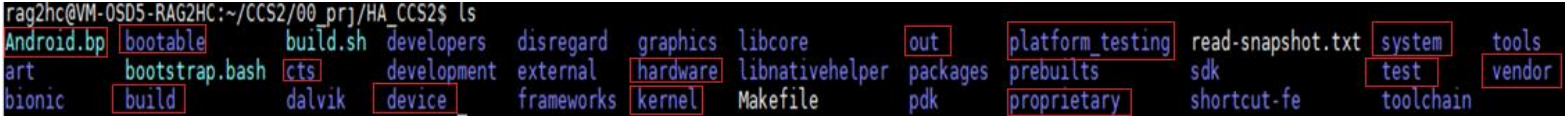


```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2$ ls
Android.bp  bootable        build.sh  developers   disregard   graphics  libcore          out       platform_testing  read-snapshot.txt  system     tools
art         bootstrap.bash  cts       development  external    hardware  libnativehelper  packages  prebuilts         sdk                test       vendor
bionic      build           dalvik    device       frameworks  kernel    Makefile         pdk       proprietary       shortcut-fe        toolchain
```

Android build system

- The build output is placed in ‘**out**’ folder. Stuff under:
  - ’**out/host**' are things compiled for your host platform (your desktop machine).
  - '**out/target/product/<platform-name>**' are things compiled for a target device or emulator.
  - '**out/target/product/<platform-name>/obj**' is used for staging "object" files, which are intermediate binary images used for building the final programs.
  - '**out/target/product/<platform-name>**’ lands in the file system of the target is stored in the directories root, system, and data. Usually, these are bundled up into image files called **system.img**, **ramdisk.img**, and **userdata.img**.
- This matches the separate file system partitions used on most Android devices.

Folder structure



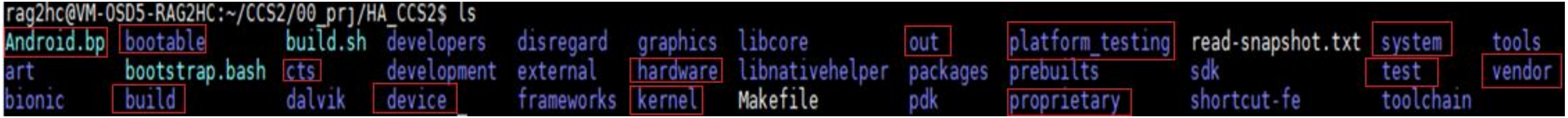
```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2$ ls
Android.bp  bootable        build.sh  developers   disregard   graphics  libcore          out       platform_testing  read-snapshot.txt  system     tools
art         bootstrap.bash  cts       development  external    hardware  libnativehelper  packages  prebuilts         sdk                test       vendor
bionic      build           dalvik    device       frameworks  kernel    Makefile         pdk       proprietary       shortcut-fe        toolchain
```

- **Android.bp**: Android blue print file. It is a new Android makefile to replace old Android makefile (Android.mk). It is a soft link to “build/soong/root.bp”

- **art**: this is the place where the Android Run Time virtual machine source is stored. It includes **Android ART runtime**, **compiler**, **dex2oat**, etc.

- **bionic**: Android C standard library

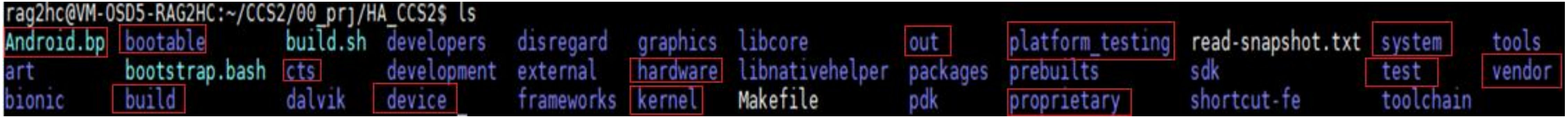
Folder structure



```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2$ ls
Android.bp  bootable        build.sh  developers   disregard   graphics  libcore          out       platform_testing  read-snapshot.txt  system     tools
art         bootstrap.bash  cts       development  external    hardware  libnativehelper  packages  prebuilts         sdk                test       vendor
bionic      build           dalvik    device       frameworks  kernel    Makefile         pdk       proprietary       shortcut-fe        toolchain
```

- **bootable:** contains code samples regarding Android device boot. In this folder, we will find the protocol used by all Android bootloaders and a recovery image

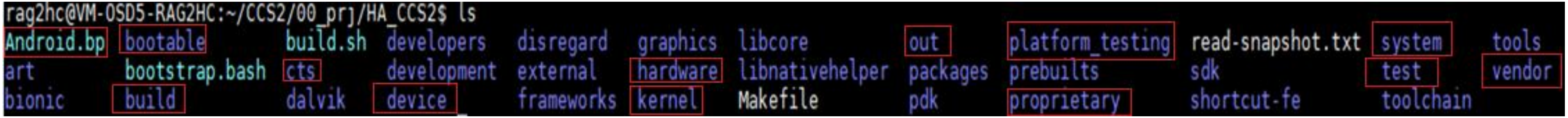
Folder structure



```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2$ ls
Android.bp  bootable        build.sh  developers   disregard   graphics  libcore          out       platform_testing  read-snapshot.txt  system     tools
art         bootstrap.bash  cts       development  external    hardware  libnativehelper  packages  prebuilts         sdk                test       vendor
bionic      build           dalvik    device       frameworks  kernel    Makefile         pdk       proprietary       shortcut-fe        toolchain
```

- **bootstrap.bash:** This script serves two purposes:
  - It can bootstrap the standalone blueprint to generate the minibp binary (by running the script with no arguments from the desired build directory).
  - It can also be invoked from another script to bootstrap a custom blueprint- based build system (the invoking script must first set some or all of the following environment variables: BOOTSTRAP, WRAPPER, SRCDIR, BLUEPRINTDIR, BUILDDIR, NINJA_BUILDDIR, GOROOT)

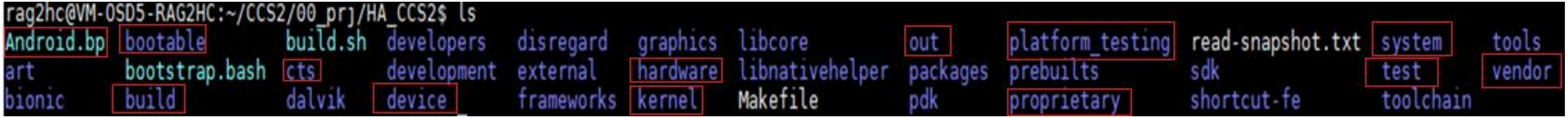
Folder structure



```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2$ ls
Android.bp  bootable        build.sh  developers   disregard   graphics  libcore          out       platform_testing  read-snapshot.txt  system     tools
art         bootstrap.bash  cts       development  external    hardware  libnativehelper  packages  prebuilts         sdk                test       vendor
bionic      build           dalvik    device       frameworks  kernel    Makefile         pdk       proprietary       shortcut-fe        toolchain
```

- **build:** contains the core components of the build system. It includes Makefile-based AOSP build system.
- **build.sh:** This is the soft link to vendor/qcom/opensource/core-utils/build/build.sh. This is product- specific.
- **cts:** Android compatibility test suite.
- **dalvik:** contains the source code of Dalvik virtual machine

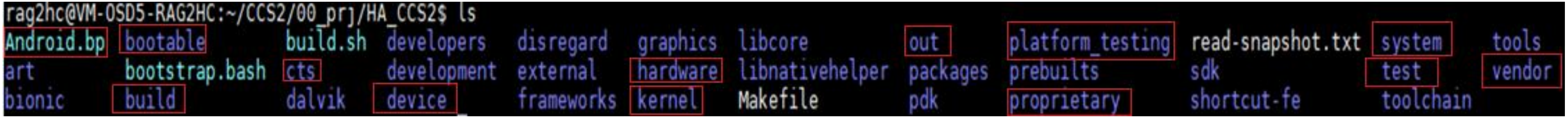
Folder structure



```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2$ ls
Android.bp  bootable        build.sh  developers   disregard   graphics  libcore          out       platform_testing  read-snapshot.txt  system     tools
art         bootstrap.bash  cts       development  external    hardware  libnativehelper  packages  prebuilts         sdk                test       vendor
bionic      build           dalvik    device       frameworks  kernel    Makefile         pdk       proprietary       shortcut-fe        toolchain
```

- **developers:** Source code example.
- **development:** Contains the development tools, debug applications, API samples, etc
- **device:** It contains the device specific configurations for many devices
- **external:** is one of the largest folders in the source code, it contains all the external projects used in the Android code

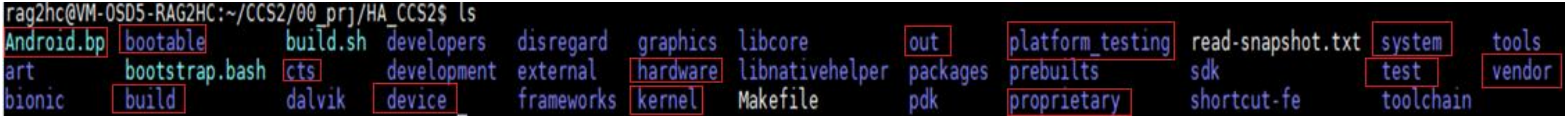
Folder structure



```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2$ ls
Android.bp  bootable        build.sh  developers   disregard   graphics  libcore          out       platform_testing  read-snapshot.txt  system     tools
art         bootstrap.bash  cts       development  external    hardware  libnativehelper  packages  prebuilts         sdk                test       vendor
bionic      build           dalvik    device       frameworks  kernel    Makefile         pdk       proprietary       shortcut-fe        toolchain
```

- **framework:** Android core components
- **graphics:** contains source code of Weston and Wayland based on Wayland protocol
- **hardware:** Android Hardware Abstraction Layer and hardware support
- **kernel:** Linux Kernel source code with Android modifications

Folder structure

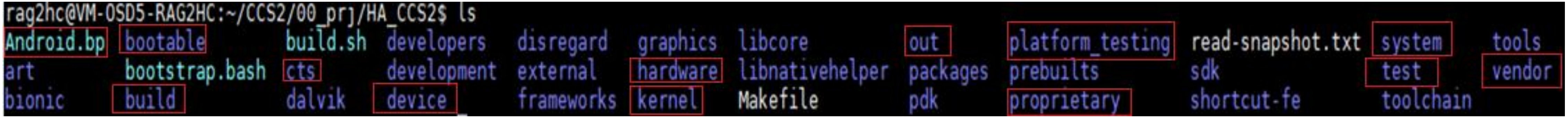


```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2$ ls
Android.bp  bootable        build.sh  developers   disregard   graphics  libcore          out       platform_testing  read-snapshot.txt  system     tools
art         bootstrap.bash  cts       development  external    hardware  libnativehelper  packages  prebuilts         sdk                test       vendor
bionic      build           dalvik    device       frameworks  kernel    Makefile         pdk       proprietary       shortcut-fe        toolchain
```

- **libcore:** Java core libraries
- **libnativehelper:** contains a few JNI helpers for the Android base classes
- **ndk:** Native Development Kit source code & tools (for using C/C++ from APKs)
- **out:** Build output directory (make clean ~= rm -rf out)

BOSCH

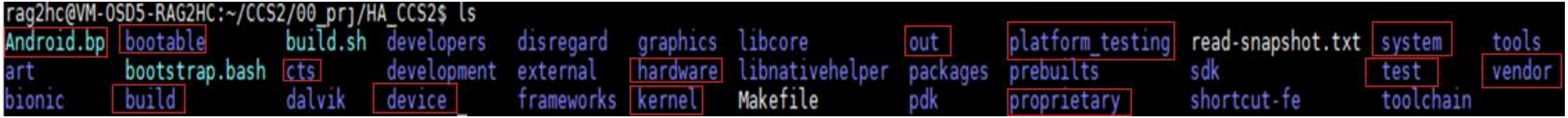
Folder structure



```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2$ ls
Android.bp  bootable        build.sh  developers   disregard   graphics  libcore          out       platform_testing  read-snapshot.txt  system     tools
art         bootstrap.bash  cts       development  external    hardware  libnativehelper  packages  prebuilts         sdk                test       vendor
bionic      build           dalvik    device       frameworks  kernel    Makefile         pdk       proprietary       shortcut-fe        toolchain
```

- **package:** contains the standard Android applications
- **pdk:** Platform Development Kit for OEMs to test upcoming AOSP versions
- **platform_testing:** contain some tools, script, libraries to test the whole android platform
- **prebuilt:** contains all the prebuilt binaries, most notably the toolchains

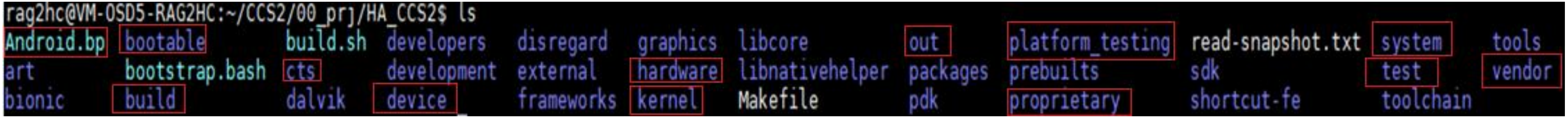
Folder structure



```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2$ ls
Android.bp  bootable        build.sh  developers   disregard   graphics  libcore          out       platform_testing  read-snapshot.txt  system     tools
art         bootstrap.bash  cts       development  external    hardware  libnativehelper  packages  prebuilts         sdk                test       vendor
bionic      build           dalvik    device       frameworks  kernel    Makefile         pdk       proprietary       shortcut-fe        toolchain
```

- **proprietary:** contains all proprietary components from vendors
- **sdk:** Android software development kit
- **shortcut-fe:** an in-Linux-kernel IP packet forwarding engine. It's designed to offer very high speed IP packet forwarding based on IP connection tracking.

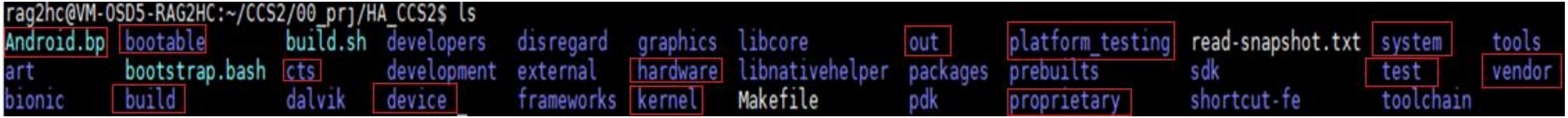
Folder structure



```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2$ ls
Android.bp  bootable        build.sh  developers   disregard   graphics  libcore          out       platform_testing  read-snapshot.txt  system     tools
art         bootstrap.bash  cts       development  external    hardware  libnativehelper  packages  prebuilts         sdk                test       vendor
bionic      build           dalvik    device       frameworks  kernel    Makefile         pdk       proprietary       shortcut-fe        toolchain
```

- **system:** contains all the basic pieces of the Android system: init, shell, the volume manager, etc.
- **test:** this is optional folder. It includes all test cases and test suites to test the whole system
- **toolchain:** this is optional folder. It can contain some external tools to check/ verify the whole system

Folder structure



```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2$ ls
Android.bp  bootable        build.sh  developers   disregard   graphics  libcore          out       platform_testing  read-snapshot.txt  system     tools
art         bootstrap.bash  cts       development  external    hardware  libnativehelper  packages  prebuilts         sdk                test       vendor
bionic      build           dalvik    device       frameworks  kernel    Makefile         pdk       proprietary       shortcut-fe        toolchain
```

- **tools:** compilation and IDE tools – gradle files, Eclipse add-ons, Studio add-ons.

- **vendor:** OEM chipset binaries and device definitions

Thank for your listening!