Android AOSP Overview

Karthik Dantu and Steve Ko

Administrivia

- Any issues in building?

Android Build System & Source Tree

- Today's goal
  - Getting to know the build system
  - Navigating the source tree
- Resources
  - <android source root>/build/core/build-system.html
  - http://elinux.org/
  - "Embedded Android" by Karim Yaghmour
  - Other sources: "Android Building" google groups

Git & Repo

- AOSP is a set of repositories, each individually managed by git.
  - If you don't know git, please go find out!
- repo is a tool that manages multiple git repositories.
- .repo directory: management files
  - It's actually another git repository.
  - repo init: just clones this repository.
  - repo sync: clones other repositories based on manifest.xml file.

Try Out!

- You might need to add your own repository for your project.

```
$ cd .repo
$ ls –al
$ cd manifests
$ vi default.xml
    • (vi or whatever editor)
$ git branch -a
```

Build Targets

- Build name + build type (eng, user, & userdebug)
- Generates different builds for different purposes
  - eng: produces an “engineering” build.
  - user: produces a release build.
  - userdebug: produces a release build plus some limited debugging support.
- Sets different system properties.
  - getprop, setprop
  - property_get(), property_set()

Build Targets

- eng
  - adb enabled and runs as root by default (ro.secure=0)
  - DDMS & debugger connection enabled (ro.debuggable=1)
- user
  - adb disabled by default & no debugging
- userdebug
  - adb enabled (does not run as root) & debugging support

Try Out!

- You might want to define system-wide properties.

```
$ adb shell getprop
$ adb shell getprop ro.secure
$ adb shell getprop ro.debuggable
```

Build Environment

- Useful commands after source-ing build/envsetup.sh
  - croot: cd to the source root.
  - m: make from the source root.
  - mm: make in the current module.
  - mmm: make what's specified.
  - cgrep, jgrep, mgrep: grep in C, Java, and Makefiles.
  - godir: go to a directory that contains a file. (If you run it, it will index first.)
  - printconfig: prints out the current configuration.

Try Out!

$ godir Android.mk

Make System

- Combines
  - Build configuration (build/envsetup.sh & lunch)
  - Core .mk files (build/core/main.mk, build/core/config.mk, build/core/definitions.mk, build/core/Makefile, etc.)
  - Module description (Android.mk) files
  - Product descriptions (AndroidProducts.mk & .mk for the specific product)
  - Board description (BoardConfig.mk) files

Make Targets

- make showcommands: shows all the make commands as it goes on. (good for debugging)
- make clobber: the cleanest removal.
- make modules: shows all modules that can be built individually.
- make <module name>: builds just that module.
- Android.mk: Makefile for a module.
  - LOCAL_MODULE is the target name for a module

Try Out!

$ make modules
$ cd dalvik/dalvikvm
$ vi Android.mk

Source Tree

- abi: C/C++ Application Binary Interface
- bionic: Android's C library
- bootable: Things related to booting (bootloader, installer, etc.)
- build: Build tools and makefiles
- cts: Compatibility Test Suite
- dalvik: Dalvik VM
- developers: Code samples

Source Tree

- development: Tools, scripts, files related to development (e.g., emulator, ndk, & sdk)
- device: Device-specific files
- docs: source.android.com docs
- external: various tools/libraries from external sources (other open source projects)
- frameworks: the Android framework
  - Especially, base & base/core (android.* & com.google.*)

Source Tree

- hardware: Hardware support libraries, e.g., Wi-Fi, framebuffers, etc.
- libcore: Core Java libraries (java.*) mostly from Apache Harmony VM
- libnativehelper: Helper functions written in C/C++
- ndk: Native Development Kit
- packages: Essential APKs that gets installed, e.g., Browser, Phone, ContactsProvider, etc.
- pdk: Platform Development Kit

Source Tree

- prebuilts: Prebuilt tools for compiling (mainly for cross-compiling)
- sdk: Software Development Kit
- system: Core system tools and libraries, e.g., adb, fastboot, sh, init, logcat, wlan, liblog, & libcutils.
- tools: Other external tools
- vendor: Vendor-specific files

File System Structure

/boot: kernel image and ramdisk

/system: the Android system

/recovery: recovery image & space for the recovery mode

/data: user data, e.g., apps (/data/app & /data/data/<app name>), contacts, etc.

/cache: space for caching (apps can also use this space.)

/misc: space for system settings

/sdcard: internal SD card (symlinked to /mnt/sdcard)

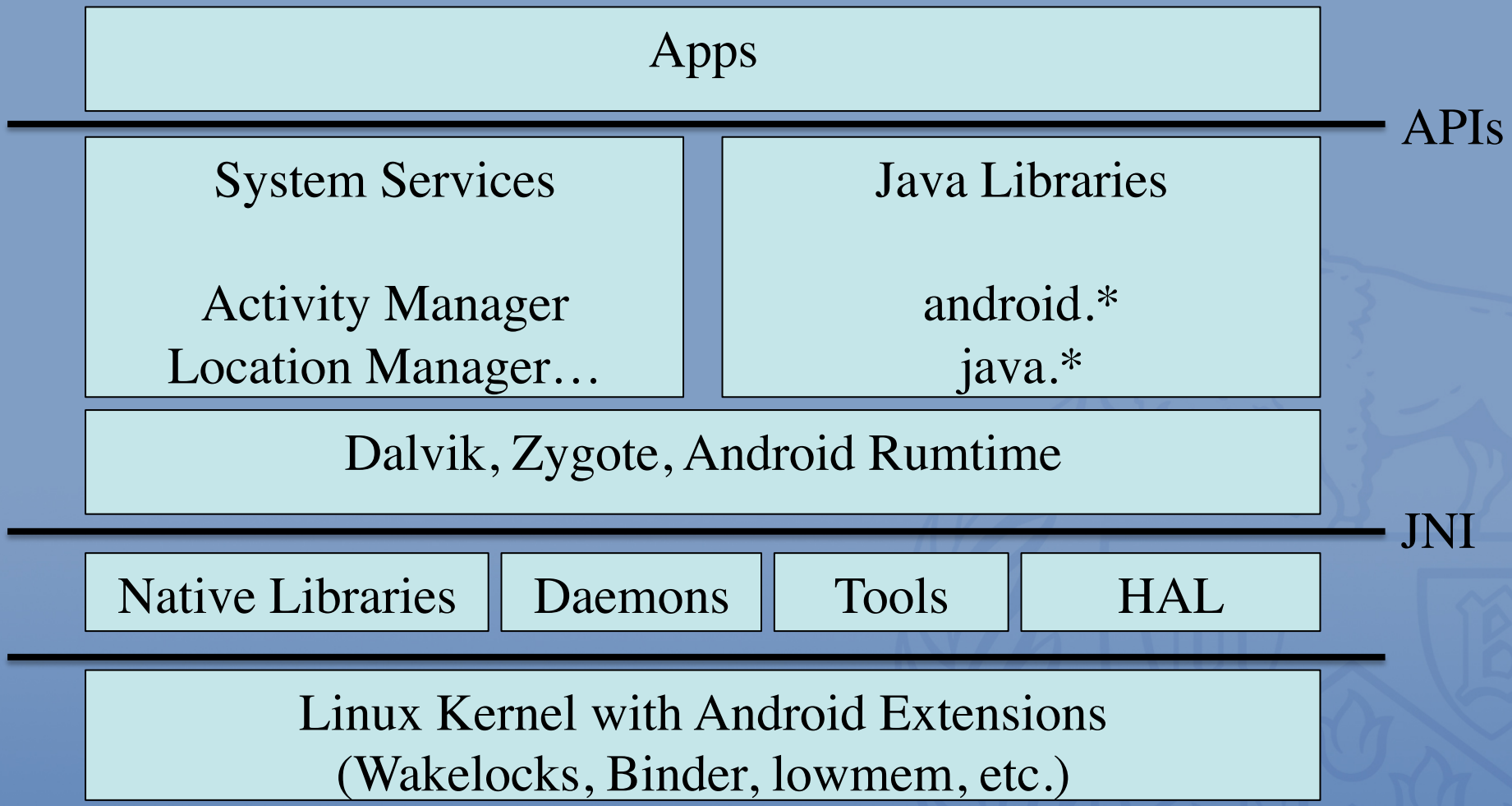
Other Pseudo File Systems

- cgroup: Linux control groups
  - /proc/cgroups
- procfs: kernel live data, e.g., parameters, processes...
  - /proc
- sysfs: kernel objects, e.g., devices, FSes, modules,...
  - /sys
- tmpfs: devices available
  - /dev

Try Out!

$ adb shell df

Architecture



Daemons

- adbd: adb daemon
- vold: volume manager daemon
- rild: radio interface layer daemon
- app_process: Zygote
- servicemanager: manager for all services

Kernel Extensions

- wakelocks
- lowmem handler
- Binder
- ashmem (Anonymous Shared Memory)
- Logger