Building Your Own Android Systems from Source

Adam C. Champion, Ph.D.
CSE 5236: Mobile App Development
Outline

• Motivation
• Preparing a Build Environment
• Obtaining Source Code
• Building Systems
Motivation

- **Why build your own system?**
  - Full control over device software (remove “bloat,” address security and privacy)
  - Extend device lifetime beyond “official” OS support
  - Research (e.g., wireless networking, mobile systems, etc.)
  - Customize device (e.g., themes, lockscreens)

- **Disclaimers:**
  - This will **definitely** void your warranty – system building is **at your own risk!** I am **not** responsible if you “brick” your device!
  - Using a custom Android system on a cellular network may violate your network operator’s terms of service
  - We’ll **only** cover building Android Open Source Project (AOSP); Google services are proprietary
Preparing a Build Environment (1)

• What you’ll need: [1]
  – 64-bit Mac OS X or Linux (Ubuntu LTS, Debian)*
  – ≥ 16 GB RAM
  – ≥ 150 GB disk space to build Android system; ≥ 200 GB for multiple builds
  – JDK version 5–7 (see table)
  – Python 2.6–2.7
  – GNU Make 3.81–3.82
  – Git
  – A few hours …

<table>
<thead>
<tr>
<th>JDK Version</th>
<th>Android Version to be Compiled</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1.5–2.2</td>
</tr>
<tr>
<td>6</td>
<td>2.3–4.4</td>
</tr>
<tr>
<td>7</td>
<td>5+</td>
</tr>
</tbody>
</table>

* MS Windows users need to create Linux VM (e.g., using VirtualBox [2]). The above system requirements are minimum for building (plus VM overhead).
Preparing a Build Environment (2)

• Install prerequisites [3]:
  – OS X: Xcode, command-line utilities (Mac App Store); GNU make, git, ccache from MacPorts, Homebrew
  – Linux: Software updates, needed packages [3]
  – ccache is strongly recommended to speed up build process [3]

• Linux users need to enable USB access: [4]
Preparing a Build Environment (3)

• Install repo, create a working directory. Run the following in a shell: [4]

$ mkdir ~/bin
$ PATH=$HOME:$PATH
$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
$ mkdir WORKING_DIRECTORY
$ cd WORKING_DIRECTORY
$ repo init -u https://android.googlesource.com/platform/manifest
$ repo sync

– Be prepared to wait a few hours, especially with the last command!
Building the System

• Run the following in a shell: [4]
  $ . build/envsetup.sh
  $ lunch
  $ # Pick an option. Make sure to select a “*-userdebug” or “*-eng” build
  $ export USE_CCACHE=1
  $ export CCACHE_DIR=/<path_of_choice>/\.ccache
  $ prebuilts/misc/linux-x86/ccache/ccache -M 50G
  $ make -j4

  On OS X, replace linux-x86 with darwin-x86

• This process takes several hours. An SSD greatly decreases compile time…
Installing System on Device

• Prerequisites:
  – Unlock device bootloader (voids warranty)
  – Get root access on your device (varies for device, voids warranty)
• Google Nexus devices (up to Nexus 6) strongly recommended for system building
• Ensure you download the proprietary binaries for your (Nexus) device from [5], device manufacturer
• Then run `tar xzvf <driver>` at a shell for each downloaded driver `<driver>` (may vary for devices)
• Now run at a shell:
  
  $ adb reboot bootloader
  $ fastboot flashall -w
More to Explore

• Build CyanogenMod (a custom Android ROM) via similar process as AOSP [6]
• Xposed: JAR file loaded into each Android executable for customization [7]
• Bluetooth: Swap Linux BlueZ stack with Android’s “BlueDroid” on Nexus devices [8]
• F-Droid: Free/open-source Android software repository alternative to Google Play [9]
• The rest is up to you!
Thank You

Questions and comments?
References

1. AOSP, https://source.android.com/source/requirements.html
2. Oracle, VirtualBox, http://www.virtualbox.org
6. CyanogenMod, https://wiki.cyanogenmod.org/w/Main_Page