Building Your Own Android Systems from Source

Instructor: 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!** Not responsible for “bricked” devices!
  – 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 macOS, Ubuntu Linux 14.04*
  – 16 GB RAM
  – 250 GB disk space for code; 150 GB for builds
  – JDK (for earlier versions)
  – Python 2.7
  – GNU Make
  – Clang/LLVM (for Android 8+)
  – Git
  – A few hours …

<table>
<thead>
<tr>
<th>JDK Version</th>
<th>Android Version</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]). Google provides Docker file to automate package installation.
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 strongly recommended (speeds up build) [3]
• Linux users need to enable USB access: [4]

```
```
Preparing a Build Environment (3)

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

  $ mkdir ~/bin
  $ PATH=~/bin:$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. SSDs greatly decreases compile time…
Installing System on Device

• Prerequisites:
  – Unlock device bootloader (**voids warranty**)
  – Get root access on your device (varies among devices; **voids warranty**)

• Ensure you download the proprietary binaries for your device from device manufacturer [5]

• Then run `tar xvzf <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 LineageOS (a custom Android ROM) via similar process as AOSP [6]
• Xposed: JAR file loaded into each Android executable for customization [7]
• F-Droid: Open-source Android software repository alternative to Google Play [8]
• 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