CSC 472 / 372 Mobile Application Development for Android

Prof. Xiaoping Jia
School of Computing, CDM
DePaul University
xjia@cdm.depaul.edu
@DePaulSWEng

Hardware Architectures and Android Emulators

Mobile Device Hardware

- Mobile devices uses different hardware (CPU/GPU) than laptop and desktop computers
  - Performance and low power consumption
- Hardware architecture design
  - Distinct instruction set for CPU/GPU
  - A family of CPU/GPU share a similar design
  - Chipsets can be designed and fabricated based on a similar design
  - Abstract Binary Interface (ABI)
- For best performance, programs are compiled to binary code using the instruction of the target architecture
- In general, programs compiled for one architecture cannot directly run on a different architecture

Mobile Device Hardware

- Intel/AMD x86 architecture dominates the laptop and desktop market
- Several distinct architectures are available for the mobile and embedded device market
  - ARM
  - x86
  - MIPS
- Most mobile devices use chipsets based on ARM (90%)
  - Nexus, Samsung, LG, etc.
  - Apple iPhone/iPad also uses ARM based chipsets

Android Hardware Support

- Android OS has been ported to
  - ARM, x86, MIPS
- 64-bit and multi-core support added since Android 5.0 (Lollipop)
- Transparent to most developers
  - Except to *native* developers using NDK (*Native Development Kit*)
- For most developers, you expect the same behavior and outcomes regardless of the architectures

ARM Processors

- ARM (*Advanced RISC Machines*) based in Cambridge, UK
- Family of RICS (reduced instruction set computing) processors used in phones/tablets/embedded devices
- Low power consumption, high performance
- Single and multi-core
- ARMv7 (32-bit) ARMv8 (64-bit)
- Chipsets based on ARM designs are manufactured by many companies, and widely used in phones and tablets
  - Including Samsung, Qualcomm, Nvidia, TI, and Apple

Intel Atom Processors

- Ultra-low power version of *Intel x86* processor family
- CISC (complex instruction set computing) processors
  - Compatible instruction set with x86 family of CPUs widely used in laptop/desktop computers
- Single and multi-core
- x86 (32-bit) and x86_64 (64-bit)
- Primarily used in netbooks and low-cost tablets
- Many x86-based devices can also run ARM binaries
  - Intel's *Binary Translator* – runtime instruction translation

MIPS Processors

MIPS TECHNOLOGIES

- *MIPS Technologies* based in Sunnyvale, CA, US
  - Founded in 1984 by researchers from Stanford University, including John L. Hennessy, President of Stanford University
- Family of RICS processors
- High performance, low power consumption
- Single and multi-core
- MIPS (32-bit) and MIPS64 (64-bit)
- Primarily used in embedded devices
  - Routers, digital TV sets, set-top boxes, and game consoles, etc.

Android Virtual Devices – Performance Issues

- AVDs are *emulators* based on QEMU
- Most mobile devices use ARM based chipsets
- Most desktops/laptops are Intel x86 systems
- Emulators are highly processor intensive
  - i.e., *slow*, to emulate a system image for ARM on an Intel x86 host machine

Emulators vs. Simulators

- Both are software that execute programs compiled for a target platform on a different host platform
  - Target platforms: Google Nexus 6, Apple iPhone 6
  - Host Platforms: Windows 10, Mac OSX 10.10
- Emulators, e.g., Android Virtual Devices (AVDs)
  - Mimic the execution of each instruction using software to produce the exact behavior, states, and results
  - Limited to the resources available on the target platform
    - Can reproduce resource related issues
- Simulators, e.g., iOS Simulators
  - Simulate the behavior and produce the same observable results by executing instructions and using resources of the host system
  - Can be faster than the real devices.

Choosing Android Emulators

- Which architecture for AVD should I choose?
  - ARM, if you want accurate emulation of a large majority of the market
  - x_86, if you want the best performance of the *emulator*
    - All architectures should produce the same outcome.
    - Good enough for most developers, unless you develop native code.
- AVD configurations for better performance
  - Use system images for x86
  - Use the host machine for graphics
  - Install *Intel x86 Emulator Accelerator* (HAXM)

Alternatives to AVD

- Use a different emulator
  - Genymotion, uses virtualization technologies on x_86 processors
- Use actual devices

Using Genymotion Emulators

GENYMOTION

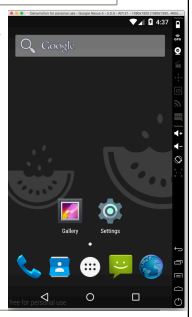
Genymotion Emulators

- A commercial version of Android emulators
  - Based on open source *AndroVM*
  - https://www.genymotion.com/
  - Free and paid versions (more features)
- Emulators using Intel x86 architecture *virtualization*
  - Oracle *Virtual Box* (https://www.virtualbox.org/)
  - Intel x86 architecture only
- Hardware acceleration of graphics (OpenGL)
  - Use host machine

DePaul University 14

Using Genymotion

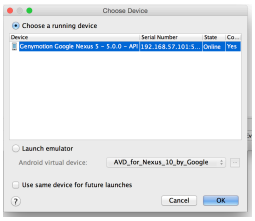
- Download and install stand-alone tool
- Start Genymotion Virtual Device Manager
- Select or add a virtual device
- Start a virtual device

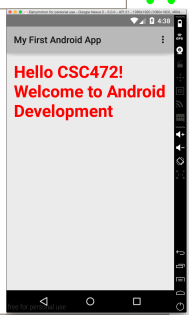


DePaul University 16

Using Genymotion

- In Android Studio, run an app
- Choose a Genymotion device

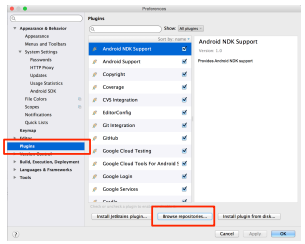




DePaul University 17

Using Genymotion Plugin

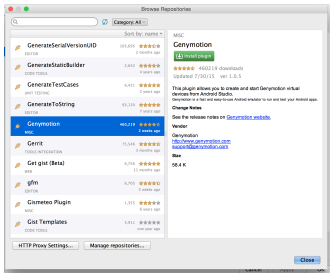
- In Android Studio, open **Preferences**
- Select *Plugins*
- Click *Browse Repositories*



DePaul University 18

Using Genymotion Plugin

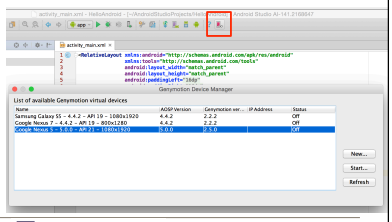
- Select *Genymotion* from the list
- Click *Install Plugin*
- Confirm and restart Android Studio



DePaul University 19

Using Genymotion Plugin

- The Genymotion icon will appear in the toolbar
- Click on the icon to launch the *Genymotion Virtual Device Manager*
- Run an app



DePaul University 20