

Introduction

Lecture 1

Operating Systems Practical

5 October 2016

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

Team, Schedule and Grading

Android Architecture

Application development overview

Team, Schedule and Grading

Android Architecture

Application development overview

- ▶ Team
 - ▶ Laura Gheorghe, Vlad Traista-Popescu, invited speakers
- ▶ Schedule
 - ▶ Lecture: Wednesday, 18-20, room PRECIS 706
 - ▶ Labs: Wednesday, 20-22, room PRECIS 706

- ▶ Exams
 - ▶ **2.5 points** Mid-term exam
 - ▶ **2.5 points** Final exam
- ▶ Labs & Project
 - ▶ **2.4 points** Lab activity
 - ▶ **3 points** Project

- ▶ Android OS:
 - ▶ SDK
 - ▶ Internals
 - ▶ Services
 - ▶ Connectivity
 - ▶ NDK
 - ▶ Native libraries
 - ▶ Security

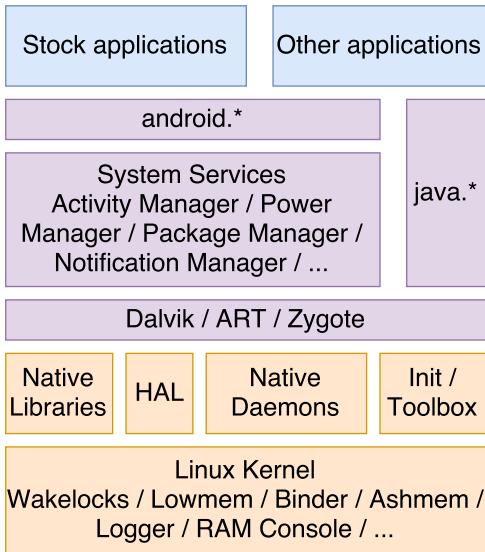
- ▶ Wiki: <http://ocw.cs.pub.ro/courses/osp>
- ▶ Moodle:
<http://cs.curs.pub.ro/2016/course/view.php?id=179>

Team, Schedule and Grading

Android Architecture

Application development overview

- ▶ Open-source OS for mobile devices
- ▶ More than 87.6% of mobile market share (2016 Q2)
- ▶ Official application market: Google Play



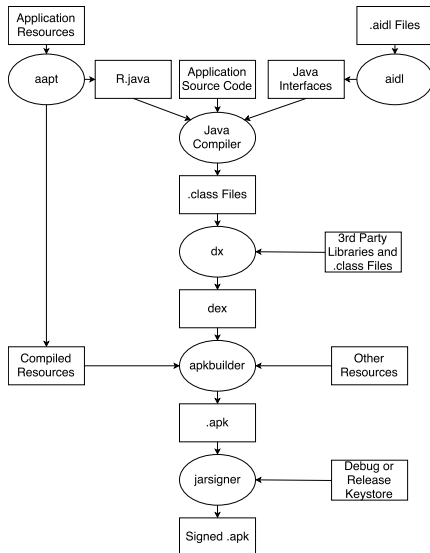
- ▶ Linux kernel
 - ▶ Android Mainlining Project / Android Upstreaming
 - ▶ Androidisms
 - ▶ Advantages
- ▶ Hardware Abstraction Layer (HAL)
 - ▶ Standard interfaces
 - ▶ Multiple library modules
- ▶ Native userspace
 - ▶ init process
 - ▶ Native daemons
 - ▶ Native libraries
 - ▶ Through Java framework APIs
 - ▶ Through Android NDK

- ▶ Android Runtime
 - ▶ Dalvik
 - ▶ ART
 - ▶ Ahead-Of-Time (AOT) compilation
- ▶ Java Runtime libraries
 - ▶ `java.*` and `javax.*`
 - ▶ Apache Harmony Project
 - ▶ Java Native Interface (JNI)

- ▶ System services
 - ▶ Fundamental features of Android
 - ▶ Native and Java code
 - ▶ Service interface
- ▶ Android framework libraries
 - ▶ Base components for app development
 - ▶ Interaction with the hardware
 - ▶ Interaction with high level services
 - ▶ Framework APIs

- ▶ Modified to work on mobile devices
- ▶ Patches on top of mainline Linux
- ▶ Android Mainlining Project / Android Upstreaming
- ▶ Wakelocks (also added to Linux 3.5)
- ▶ Low-Memory Killer (3.10)
- ▶ Binder (3.19)
- ▶ Alarm (3.20)
- ▶ Logger (3.20)
- ▶ Only suspend to memory

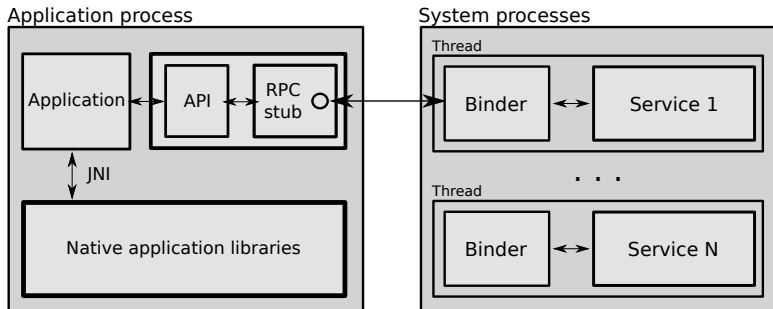
- ▶ Default until Android 5.0
- ▶ Runs Dalvik-specific byte-code
- ▶ Dalvik Executable Format (DEX)
 - ▶ Runs .dex files instead of .jar files
 - ▶ .dex is 50% smaller than corresponding .jar
- ▶ Just-In-Time compilation
 - ▶ From Android 2.2
 - ▶ Short segments of bytecode translated into native machine code at runtime
 - ▶ Improves performance



- ▶ From Android 5.0
- ▶ More advanced runtime architecture
- ▶ Ahead-Of-Time compilation
 - ▶ Just once, at installation
 - ▶ Entire DEX file -> executable for target device
 - ▶ Instead of JIT compilation and Dalvik interpretation
 - ▶ More efficient, reduced power consumption
 - ▶ More space to store the executables
- ▶ Improved memory allocation, GC, debugging and profiling

- ▶ bionic (libc)
 - ▶ Much smaller and faster than glibc
- ▶ SQLite
 - ▶ Managing SQL databases
- ▶ OpenGL ES
 - ▶ Standard software interface for 3D processing hardware
- ▶ WebKit
 - ▶ Display web pages
 - ▶ Android, Apple iOS, BlackBerry, Tizen
- ▶ SSL
 - ▶ Securing the communication over Internet

- ▶ System Services and Managers
 - ▶ Telephony
 - ▶ Location
 - ▶ Activity
 - ▶ Package
 - ▶ Notification
- ▶ System Content Providers
 - ▶ Calendar
 - ▶ Dictionary
 - ▶ Contacts
 - ▶ Settings



Team, Schedule and Grading

Android Architecture

Application development overview

- ▶ User interaction
 - ▶ Activities
- ▶ Background functionality
 - ▶ Services
 - ▶ Broadcast Receivers
 - ▶ Content Providers

- ▶ User interface
- ▶ Similar to a window from window-based GUIs
- ▶ User interacts with a single activity at a time
- ▶ Activities stack
 - ▶ Activities launch other activities
 - ▶ Back button for returning to the previous activity
 - ▶ No forward button
- ▶ Start activities with intents
 - ▶ Launcher or application components

- ▶ Performs operations in background, no UI
- ▶ Runs in the same process as the application
 - ▶ Can be configured to run in another process
- ▶ Tasks for the current application
- ▶ Provide services to the other applications
- ▶ Communication through the Binder

- ▶ Receive broadcast announcements
 - ▶ Low battery
 - ▶ Reboot
 - ▶ Application state changes
- ▶ Can receive global or local broadcasts
- ▶ Choose which broadcasts to receive
 - ▶ Intent filters
- ▶ Intents
- ▶ Active only when receiving a broadcast

- ▶ Manage access to a structured set of data
- ▶ Required for sharing data with other apps
- ▶ File data or "structured" data
- ▶ URI: provider and table
- ▶ Content Resolver uses the URI to send a query to the provider
- ▶ Active only when responding to a request

- ▶ Similar to signals
- ▶ Send message, determine action execution
- ▶ Purpose
 - ▶ Start activities, start or bind services
 - ▶ Delivering broadcast messages to receivers
- ▶ Dispatched by the Android system
- ▶ Includes action and data
 - ▶ *ACTION_DIAL content://contacts/people/1*
- ▶ Types
 - ▶ Explicit: directed towards a specific receiver
 - ▶ Implicit: a receiver which can resolve the action

- ▶ Lightweight RPC
- ▶ Remote object invocation
- ▶ In process and interprocess
- ▶ Transmit parcels of data
- ▶ Synchronous calls (blocking)

- ▶ Karim Yaghmour, Embedded Android: Porting, Extending, and Customizing, Chapter 2
- ▶ <https://wiki.linaro.org/LMG/Kernel/Upstreaming>
- ▶ http://elinux.org/Android_Kernel_Features
- ▶ <https://www.google.com/events/io/io14videos/b750c8da-aebe-e311-b297-00155d5066d7>
- ▶ <http://developer.android.com/guide/components/activities.html>
- ▶ <http://developer.android.com/guide/components/services.html>
- ▶ <http://developer.android.com/guide/topics/providers/content-providers.html>
- ▶ <http://developer.android.com/guide/components/intents-filters.html>

- ▶ Linux kernel
- ▶ Android Runtime
- ▶ Dalvik
- ▶ ART
- ▶ Native libraries
- ▶ Application framework
- ▶ Activities
- ▶ Services
- ▶ Broadcast receivers
- ▶ Content providers
- ▶ Intents
- ▶ Binder