

Sviluppo Applicazioni Mobili

Federico Matteoni

A.A. 2019/20

Indice

1	Programmazione Android	5
1.1	Breve Storia di Android	5
1.2	Ant e Gradle	5
1.3	Architettura Android Studio/Gradle	5
2	Architettura Android	7
2.1	Struttura	7
2.2	Dalvik	7
2.3	ART	7

Introduzione

Vincenzo Gervasi, gervasi@di.unipi.it
circe.di.unipi.it/~gervasi/main/
developer.android.com

Modalità d'esame Sviluppo di un'app, proposta dallo studente ma concordata con il docente. Presentazione dell'app con ispezione del codice e domande "teoriche" su aspetti non coperti dal progetto. No compiti in classe.
3 criteri: applicazione mobile, non deve avere senso su applicazione web o su computer. diversità, almeno tre framework presentati. progetto adeguato a esame di 6 CFU.

Capitolo 1

Programmazione Android

1.1 Breve Storia di Android

2007 – telefonini Nokia

fino al 2007 ampio casino con windows ce (interfaccia pessima e tantissime versioni) – Novembre 2007 Open Platform for Mobile Handset 5 novembre – 12 novembre rilasciano Android. Chi c'era dietro: Google, eBay, China Mobile, HTC, Intel, LG, Motorola, NTT DoCoMo, Qualcomm, nVidia, Samsung, Sprint Nextel, Telecom Italia, Telefónica, Texas Instrument, T-Mobile. Produttori di telefoni, di chip, fornitori di servizi e di telefonia.

Rilasciato su licenza Apache, basato su Linux 2.6 e sviluppato su Eclipse, Java e Python.

Sviluppato da Android Inc., startup californiana. Nata nel 2003 a Palo Alto, acquistata da Google nel 2005 e brevetti registrati nel 2007.

Fondata da **Andy Rubin**.

1.2 Ant e Gradle

Ant antiquato

Gradle più moderno.

Gradle Sistema di build avanzato, configurabile. Distribuito nel senso di risorse per lo sviluppo sparse in rete tramite URL. Fill-in del manifest (manifest contiene metadati per il s.o.), gradle genera e mantiene aggiornato il manifest.

`compileSdkVersion` per fill-in manifest e `buildToolsVersion` per scaricare tools se non presenti.

Lint analisi statica di codice per warnings e errori sintattici della scrittura del codice.

1.3 Architettura Android Studio/Gradle

IntelliJ con plugin: android plugin, android designer, android gradle adapter.

Si appoggia all'android SDK e a Gradle (tool separato, con plugin android e anch'esso collegato all'SDK)

Inoltre c'è il progetto, con `.properties` con config per l'ambiente di sviluppo (come dove si trova il compilatore ecc.) e `build.gradle`.

Capitolo 2

Architettura Android

2.1 Struttura

Kernel Linux Kernel Linux standard, senza fare dei derivati. Adattamenti per telefono eseguiti tramite moduli del kernel. Come display driver, driver per la tastiera keypad, driver camera, wifi, flash, audio, driver binder (IPC) che cura inter-process communication (diversamente da socket e FIFO, che non andavano bene per Android. Su android non ci sono solo file, ma oggetti con metodi, e FIFO/socket adatte per trasferire flussi di byte. Binder fa comunicare processi in termini object-oriented). Power management driver.

Librerie Librerie `.so`, che fanno tantissime cose. Tra cui surface manager (equivalente dei window system X o wayland), OpenGL ES per la grafica 3D, FreeType, SSL (HTTPS e Secure Socket Layer), WebKit, SQLite, libc (scanf, strlen...).

Android Runtime In aggiunta alle librerie, Macchina Virtuale che esegue codice applicazioni Android (Dalvik), insieme alle core libraries (garbage collector, heap, memoria...)

Separazione tra mondo Java e mondo del codice eseguibile ARM Da qui in giù c'è Java, sopra è C.

Application Framework S.O. rappresentato da oggetti nello heap. Librerie: package manager, telephony manager, activity manager, window manager, location manager (GPS), notification manager

Applicazioni Tra cui servizi interni: home, contatti, telefono, browser... Firmate a chiave asimmetrica. Platform key per applicazioni che usano funzioni critiche, chiave che firma il kernel.

2.2 Dalvik

2.3 ART

Android RunTime