Mobile OS

Symbian

BlackBerry

iOS

Window mobile

Android
<table>
<thead>
<tr>
<th>OS</th>
<th>First release</th>
<th>Last release</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>Android 1.0 September 2008</td>
<td>Android 4.0 May 2011</td>
<td>Open Handset Alliance</td>
</tr>
<tr>
<td>BlackBerry</td>
<td>BlackBerry 1.0 January 1999</td>
<td>BlackBerry 7.0 May 2011</td>
<td>Research In Motion</td>
</tr>
<tr>
<td>iOS</td>
<td>iOS 1.x June 2007</td>
<td>iOS 5.x June 2011</td>
<td>Apple</td>
</tr>
<tr>
<td>Symbian</td>
<td>Symbian 1 October 2008</td>
<td>Symbian Belle August 2011</td>
<td>Nokia (Accenture)</td>
</tr>
<tr>
<td>Windows Mobile</td>
<td>Pocket PC 2000 April 2000</td>
<td>Windows Phone October 2010</td>
<td>Microsoft</td>
</tr>
</tbody>
</table>
Some history

- Aug/2005: Google acquire Android Inc.
- Nov/2007: OHA (Open Handset Alliance)
  - TelecomItalia, Vodafone
  - Acer, Asus, Dell, Samsung, HTC, Toshiba etc.
  - Android Open Source Project (AOSP)
- Sep/2008: release of Android 1.0
- May/2011: release of Android 4.0
What is ANDROID?

• Operating System for mobile devices

• Kernel based on the Linux kernel

• Middleware, libraries and APIs written in C

• Application framework includes Java-compatible libraries

• Dalvik Virtual Machine to run Dalvik dex-code
Why ANDROID?

• A simple and powerful SDK
• No licensing, distribution, or development fees
• Development over many platform
  • Linux, Mac OS, Windows
• Excellent documentation
• Large developer community
• No constraints on new Apps.
Privacy, allarme smartphone-spia un software registra le nostre vite

Dopo il caso che ha colpito i melafonini, scoperto anche sui sistemi Android un software capace di registrare le attività e la posizione degli utenti. E c'è già chi accusa anche Nokia e Blackberry. Allarme privacy anche in due grandi centri commerciali Usa.

di LAURA BONASERA

ANCORA UNA VOLTA, gli smartphone rischiano di essere una minaccia alla privacy. In tasca o in borsa, accompagnano milioni di persone in tutto il mondo ogni giorno. E non importa se, a maneggiarlo, non sono solo gli adulti ma anche bambini e adolescenti. Dopo il caso del 'software spia' su iPhone, ora arriva quello su Android. Il sistema di Google, per essere chiari.
Anche su Mac tentativi di virus
Tsunami ci prova con la mela


ANCHE per gli utenti Apple è tempo di pensare alla sicurezza. Il centro ricerche di Eset, i produttori dell'antivirus Nod32, ha appena isolato una nuova minaccia originariamente creata per Linux, chiamata Linux/Tsunami, ora rilevata anche sui sistemi Mac, e quindi identificata come OsX/Tsunami.

Per ora della minaccia virale è stata solo scoperta l'esistenza e non risulta diffusione esterna. Ma è abbastanza per invitare anche gli utenti Mac a proteggere i propri sistemi.
ANDROID architecture

Applications
- Home
- Contacts
- Phone
- Browser
- ...

Application Framework
- Activity Manager
- Window Manager
- Content Providers
- View System
- Notification Manager
- Package Manager
- Telephony Manager
- Resource Manager
- Location Manager
- XMPP Service

Libraries
- Surface Manager
- Media Framework
- SQLite
- OpenGL|ES
- FreeType
- WebKit
- SGL
- SSL
- libc

Android Runtime
- Core Libraries
- Dalvik Virtual Machine

Linux Kernel
- Display Driver
- Camera Driver
- Bluetooth Driver
- Flash Memory Driver
- Binder (IPC) Driver
- USB Driver
- Keypad Driver
- WiFi Driver
- Audio Drivers
- Power Management
To simplify development Google provides the Android Development Tools (ADT) for Eclipse.

The applications are written in Java.

Android applications are packed into an .apk (Android Package) file by the program aapt (Android Asset Packaging Tool).

The Android NDK is a companion tool to the Android SDK that lets you build portions of your apps in native code (C/C++).
Android uses a special virtual machine, the Dalvik Virtual Machine.

Dalvik uses special bytecode. Therefore, you cannot run standard Java bytecode on Android.

Android provides a tool `dx` which allows to convert Java Class files into dex (Dalvik Executable) files.

The ADT performs automatically the conversion from class to dex files and creates the `apk` during deployment.
- Implementation of the Android virtual machine
- Test and debug your android applications.
ANDROID Tools > Android Virtual Device Manager

- Create and monitor the Virtual Machines

Windows > Android SDK and AVD Manager
- Monitor and Control the Dalvik virtual machines
- Logcat (see logged msgs)

Windows > Perspectives > DDMS
ANDROID Tools > Android Debug Bridge

- Manage the state of an emulator or device
- Run shell commands on a device
- Manage port forwarding on an emulator or device
- Copy files to/from an emulator or device

root@domnoja:/usr/local/sharesw/android-sdk-linux_86/platform-tools# ./adb

Android Debug Bridge version 1.0.29

-d
- directs command to the only connected USB device
- returns an error if more than one USB device is present.

-e
- directs command to the only running emulator.
- returns an error if more than one emulator is running.

-s <serial number>
- directs command to the USB device or emulator with
- the given serial number. Overrides ANDROID_SERIAL
- environment variable.

-p <product name or path>
- simple product name like 'sooner', or
- a relative/absolute path to a product
- out directory like 'out/target/product/sooner'.
- If -p is not specified, the ANDROID_PRODUCT_OUT
- environment variable is used, which must
- be an absolute path.

devices
connect <host>[:<port>]
- list all connected devices
- connect to a device via TCP/IP
Port 5555 is used by default if no port number is specified.

disconnect [host>[:<port>]]
- disconnect from a TCP/IP device.
Port 5555 is used by default if no port number is specified.
Using this command with no additional arguments
will disconnect from all connected TCP/IP devices.
Component > Activity

- Is a single application entity that is used to perform actions.

- An application may have many separate activities, but the user interacts with them one at a time.

- Is not required to have a user interface.

- Are divided in three categories:
  - Foreground Activity: suspended when invisible
  - Background Service: Little interaction
  - Intermittent Activity
Component > Activity

```java
public class Activity extends ApplicationContext {
    protected void onCreate(Bundle savedInstanceState) {
    }
    protected void onStart() {
    }
    protected void onRestart() {
    }
    protected void onResume() {
    }
    protected void onPause() {
    }
    protected void onStop() {
    }
    protected void onDestroy() {
    }
}
```
User interface of an Activity.

Is built with:

- **Widget classes:**
  - Layout: linear, grid, tab, list, etc.
  - TextView, EditText, Button, Form, TimePicker, etc.
  - AutoCompletition, MapView, WebView etc.

- **Menu**
Component > View > Layout

Layout: linear, grid, tab, list, etc.
Component > View > Widgets

TextView, EditText, Button, Form, TimePicker, etc.
Component > View > Menu

- First Menu Item
- Second Menu Item
- Third Menu Item
- Fourth Menu Item
- Fifth Menu Item

More
Are asynchronous messages which allow the application to request functionality from other services or activities.

An application can call directly a service or activity (explicit intent) or ask the Android system for registered services and applications for an intent (implicit intents).
Perform background tasks without providing an UI. They can notify the user via the notification framework in Android.

- Service class
  - public class MyService extends Service
  - public void onStart() {...}

- Control
  - startService()
  - stopService()

- Communication
  - Bind service with activity: use public method and properties
  - Intent
Provides data to applications, via a content provider your application can share data with other applications. Android contains a SQLite DB which can serve as data provider.
Resources

➢ **src**: project source and business logic;

➢ **gen**: auto generated file

➢ **drawableX**: images and external sources. Can be specific for different image quality;

➢ **layout**: XML layout source code;

➢ **menu**: XML menu source code;

➢ **values**: XML range value for new predefined types;
Resources (2)

- **bin/**: contains the final .apk file and other compiled resources.
- **jni/**: contains native code sources developed using the Android NDK
- **assets/**: this is empty, can be used to store raw asset files.
- **res/anim**: store the animation files.
- **res/color**: For XML files that describe colors.
- **res/xml**: for miscellaneous XML files that configure application components;
- **libs/**: contains private libraries.
➢ **AndroidManifest.xml**: The control file that describes the nature of the application and each of its components.

➢ **project.properties**: contains project settings, such as the build target.

➢ **local.properties**: customizable computer-specific properties for the build system.

➢ **ant.properties**: customizable properties for the build system.

➢ **build.xml**: the Ant build file for your project. This is only applicable for projects that you build with Ant.
Component > Manifest

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="em.home"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-permission android:name="android.permissionINTERNET" />
    <uses-permission android:name="android.permission.RUN_INSTRUMENTATION" />
    <!-<uses-permission android:name="android.permission.READ_PHONE_STATE" -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <application android:icon="@drawable/icon" android:label="@string/app_name" android:debuggable="true"
        android:name="JEmergencyApp">
        <service android:name="jade.android.MicroRuntimeService" />
        <activity android:label="@string/app_name" android:screenOrientation="portrait"
            android:theme="@android:style/Theme.NoTitleBar" android:name="HomeActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="DeviceActivity" android:screenOrientation="portrait"></activity>
        <activity android:name="HealthActivity" android:screenOrientation="portrait"></activity>
        <activity android:name="PersonalActivity" android:screenOrientation="portrait"></activity>
        <activity android:name="SplashScreenActivity" android:launchMode="singleInstance"
            android:configChanges="keyboard|orientation" android:screenOrientation="portrait"></activity>
    </application>
</manifest>
```
project source code
Activity
Service
ContentProvider

```java
package em.home;

import java.io.File;

public class DeviceActivity extends Activity{

/**
 * Instance of Jade Logger, for debugging purpose.
 */
private final Logger myLogger = Logger.getMyLogger(this.getClass().getName());

// Path del file da salvare
private final static String FILE_PATH = "ecgtrace_received.jpg";

EditText et_ecg_frequency;
EditText et_ecg_cathegory;
EditText et_oximetre_frequency;
EditText et_oximetre_percentage;
```
R.java is a generated class which contains references to resources of the res folder in the project. These resources are defined in the res directory and can be values, menus, layouts, icons or pictures or animations. For example a resource can be an image or an XML file which defines strings.
.xml: instruction file about the use of the images in different condition;
.png/.jpg/...: image file;
res\layout

<?xml version="1.0" encoding="utf-8"?>
<!-- layout principale -->
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="wrap_content">

<!-- layout secondario per la organizzazione a tabella dei textView e textedit -->
<TableLayout android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:id="@+id/top_table">

<TableRow android:layout_width="wrap_content"
android:layout_height="wrap_content" android:id="@+id/firstRow">
    <TextView android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_weight="1" android:id="@+id/name" />
    <EditText android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:text="@string/name" android:id="@+id/tname" />
</TableRow>

</TableLayout>
</ScrollView>
Android Menu

Menu Elements

- send (Item)
- connect (Item)
- settings (Item)
- receive (Item)
- exit (Item)

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/send"
        android:title="Send Data"
        android:icon="@drawable/send" />
    <item android:id="@+id/connect"
        android:title="Connect"
        android:icon="@drawable/connect" />
</menu>
Android Resources (default)

Resources Elements

- months (String Array)
- Item
- Item
- Item
- Item
- Item
- Item
- Item
- Item
- Item
- Item
- days (String Array)
- years (String Array)
- disability (String Array)
- burn (String Array)
- locprov (String Array)

Attributes for Item

- A string value to use in this string array.
- Value* 1

<!-- in questo file xml vengono contenuti tutti
<resources>
  <string-array name="months">
    <item>1</item>
    <item>2</item>
    <item>3</item>
    <item>4</item>
    <item>5</item>
    <item>6</item>
    <item>7</item>
    <item>8</item>
    <item>9</item>
    <item>10</item>
    <item>11</item>
    <item>12</item>
  </string-array>
</resources>
The development process > Setup

Set up the development environment

Install:
- Android SDK
- Android Development Tool
- Android Platforms

Set up AVD and devices for testing

Create Android Virtual Devices and connect hardware services that will be used for testing.
Create you application

Create Android project with you source code, resource files, and Android manifest file.
Build and run your application in debug mode.

Debug your application using the Android debugging and logging tools.

Test your application using the Android testing and instrumental framework.
The development process > Publishing

- Prepare your application for release
- Configure, build and test your application in release mode.
- Release your application
- Distribute your application to users.
To release an application means to make available the "apk" file.

Applications can be released in:
- Android Market
- Personal website
- Mail / p2p system, etc.
Before you release an application you must:

- Choose an application Icon
- Prepare the End-user License Agreement (EULA)
- Turn off logging and debugging
- Clean up your project directories
- Review and update your manifest settings

if release on Android Market

- Version you application
- Sign you application with an electronic certificate
http://android-tutorial.com/
http://www.androidtutorials.org/
http://developer.android.com/
http://www.vogella.de/android.html
Thank you!

For more info:

Mail: elton.domnori@unimore.it

Web: www.agentgroup.unimore.it