

Google

# Io, android

Il cellulare 2.0 diventa più personale



*Ing. Antonio Tringali*  
<[antonio.tringali@alice.it](mailto:antonio.tringali@alice.it)>  
29/03/2008



## Cosa è Android?

- Il 5/11/2007 Google annuncia Android
- E' un application environment per cellulari evoluti (*smartphone*)
- Basato su kernel Linux 2.6 (HAL) 
- Varie librerie a corredo con licenza soprattutto BSD e Apache 
- Tutti gli sviluppi originali di Android saranno rilasciati con licenza Apache 2.0 
- Prima versione non beta nel Q3 2008



# IO, ANDROID open handset alliance

- Contestualmente annunciata la OHA
- Rivolta alla commercializzazione di prodotti/servizi Android-based



Operatori Mobili	Compagnie Semiconduttori	Fabbricanti Cellulari	Compagnie Sviluppo Software	Operatori Commerciali
China Mobile KDDI NTT DoCoMo Sprint Nextel T-Mobile Telecom Italia Telefonica	Audience Broadcom Intel Marvell NVIDIA Qualcomm SiRF Technology Synaptics Texas Instruments	HTC LG Electronics Motorola Samsung Electronics	Ascender eBay Esmertec Google LivingImage LiveWire Mobile Nuance Communications PacketVideo (PV) SkyPop SONiVOX	Aplix Noser Engineering TAT Wind River



## La mente di Android

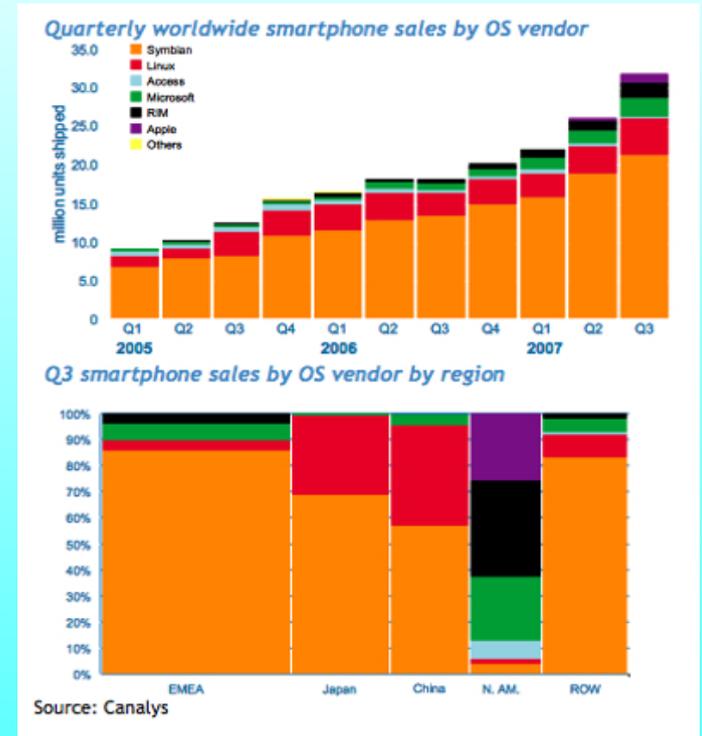
- **Andy Rubin:** tra i creatori del SW alla base del Sidekick/Hiptop
- Ha lavorato per Apple e General Magic
- Ha fondato Artemis Research → WebTV
- Ha fondato Danger, Inc. nel 2000
- Dirigenti di Apple e Philips nel consiglio
- 2004: Brin e Page lo prendono in Google
- Ora *“Director of Mobile Platforms”*
- L'11/02/2008 Microsoft compra Danger





## Una strategia convergente

- Apple iPhone: il 27% del mercato USA in **6 mesi!**
- Eric Schmidt, Google CEO, in consiglio di amm. Apple
- Nokia, Palm e RIM: i big dell'HW cellulare
- I concorrenti erano principalmente Microsoft, Symbian e Access (Linux)
- Ma Linux occupa una nicchia di mercato





## + **Guadagno** = - **Costi** + **Servizi**

- L'HW cellulare è ormai *commodity*
- Incidono i costi di licenza SW: ad esempio, Symbian e Windows Mobile
- Il mercato della telefonia è redditizio: oltre il 30% di guadagno rispetto a un 20% di costi di marketing
- Obiettivo: ricreare il mercato dei PC anni '90 in ambito cellulare
- *Subsidizing* HW per accesso a Internet



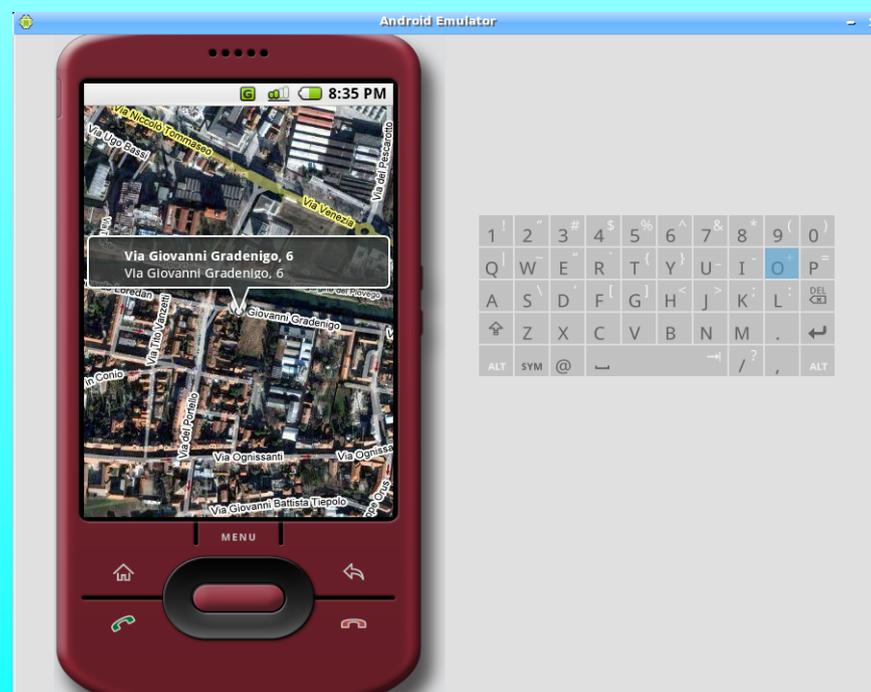
## Cosa è stato rilasciato?

- Il 12/11/2007 è stato rilasciato un SDK per “allenarsi” a produrre applicazioni
- Un plug-in Eclipse semplifica lo sviluppo

m3-rc22a (14/11/2007)



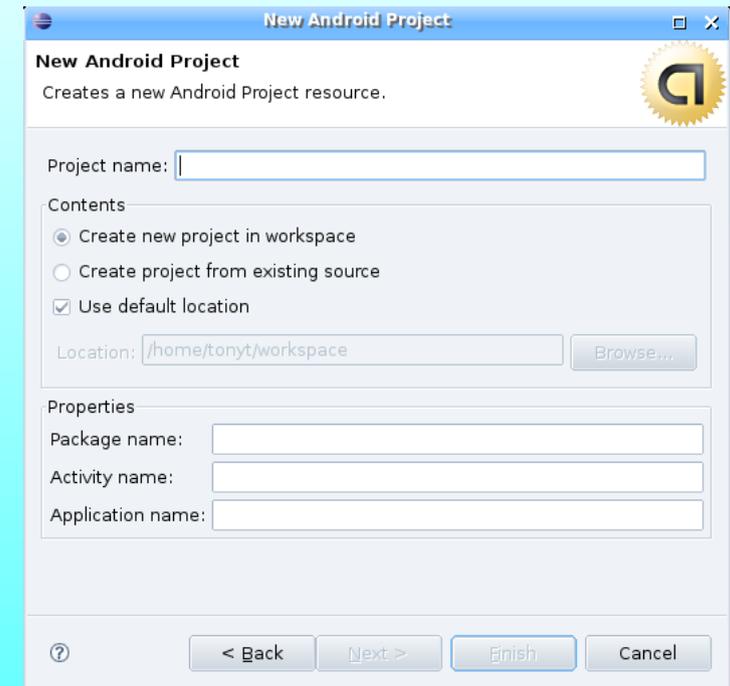
m5-rc15 (03/03/2008)





## Requisiti per l'SDK

- Rilasciato in versioni per:
  - Windows XP o Vista
  - Mac OS X  $\geq$  10.4.8 (Intel)
  - Linux Ubuntu
- Plug-in Eclipse 3.2 o 3.3
- Java Development Kit 5 o 6
- Apache ANT versione  $\geq$  1.6.5 (Linux) o 1.7.0 (Windows e Mac OS X)





## Android senza filtro

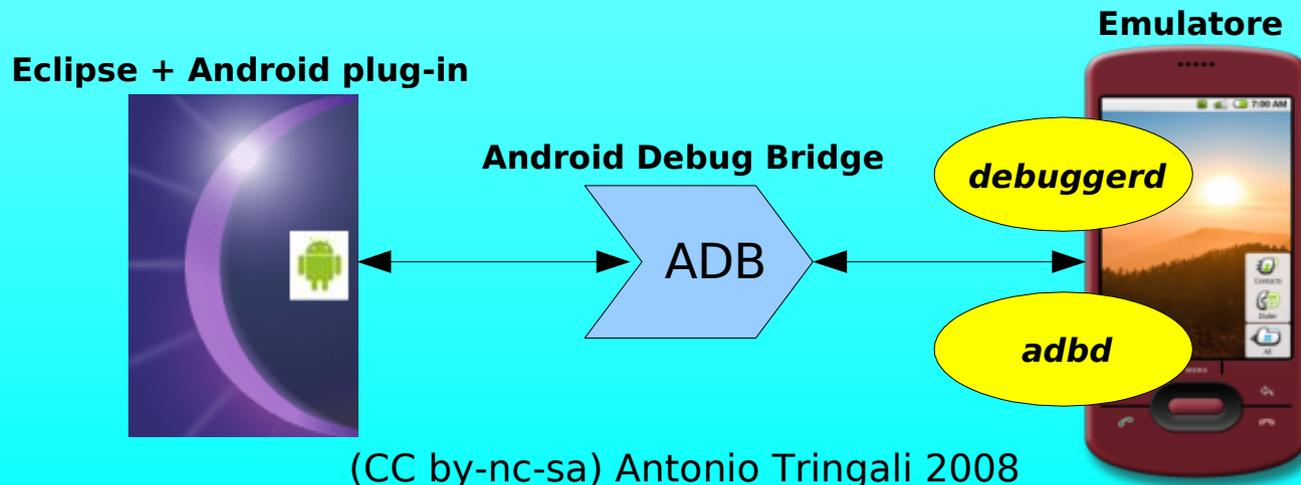
- **aapt**: crea/aggiorna .zip, .jar, .apk
- **activityCreator.py**: crea un progetto
- **dx**: .class/.jar → .dex per Dalvik
- **aidl**: genera codice interfaccia IPC
- **ddms**: debug processi emulatore
- **traceview, dmtracedump**: visual. grafica log
- **mksdcard**: immagini FAT32 per SD card
- **sqlite3**: SQLite per manipolare i database

*Eclipse plug-in  
versione 0.4.0*



## Emulazione, che passione!

- **ADB**: daemon che si interfaccia con un server **adbd** dentro il device o emulatore
- L'emulatore dispone le porte TCP/IP: 5554 per console, 5555 per protocollo ADB
- Ogni nuova istanza di emulatore alloca la coppia di porte TCP/IP successiva





# Android on steroids (1)

- Processore da 200MHz, 32MB di RAM e 32MB di flash; vari dispositivi di I/O

```
# cat /proc/cpuinfo
Processor       : ARM926EJ-S rev 5 (v5l)
BogoMIPS       : 321.12
Features        : swp half thumb fastmult vfp edsp java
CPU implementer : 0x41
CPU architecture: 5TEJ
CPU variant     : 0x0
CPU part        : 0x926
CPU revision    : 5
Cache type      : write-through
Cache clean     : not required
Cache lockdown  : not supported
Cache format    : Harvard
I size          : 4096
I assoc         : 4
I line length   : 32
I sets          : 32
D size          : 65536
D assoc         : 4
D line length   : 32
D sets          : 512

Hardware        : Goldfish
Revision        : 0000
Serial          : 0000000000000000
```

Set di istruzioni supportati:  
ARMv5TEJ, Thumb e Java (Jazelle)

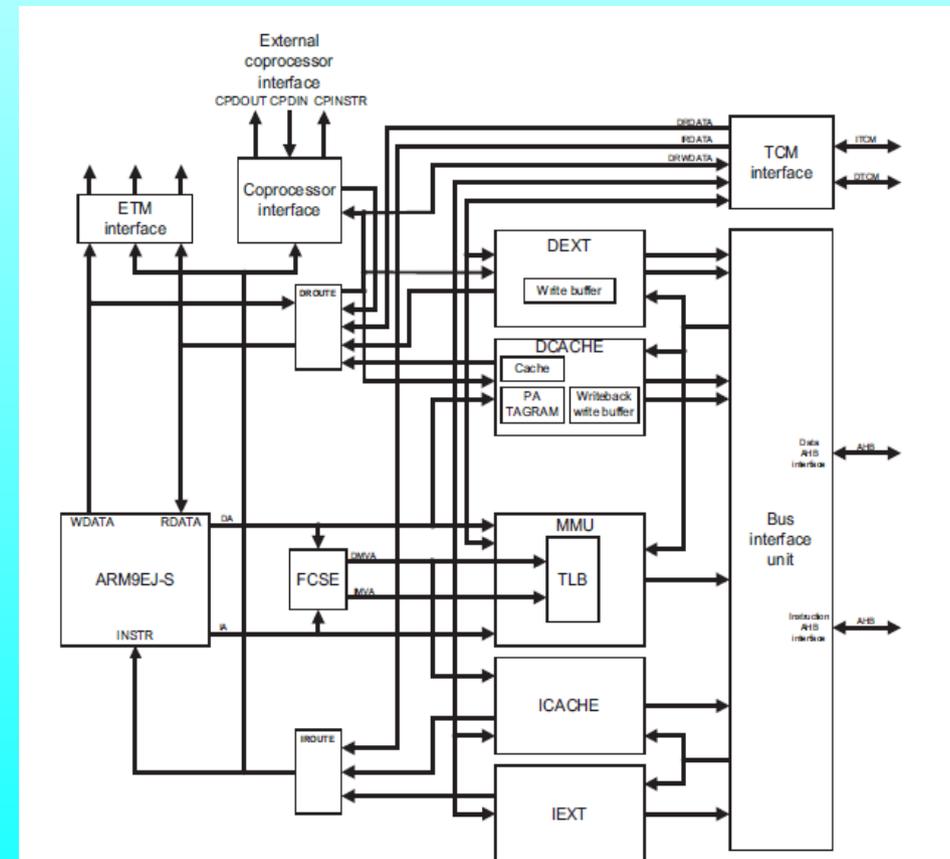


Figure 1-1 ARM926EJ-S block diagram



# Android on steroids (2)

- Vediamo in dettaglio l'HW emulato:

```
# dmesg
<5>Linux version 2.6.23-g0774dffb (arve@arvelnx.corp.google.com) (gcc
version 4.2.1) #76 Tue Jan 22 19:53:32 PST 2008
<4>CPU: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=00003137
<4>Machine: Goldfish
<4>Memory policy: ECC disabled, Data cache writeback
```



```
<5>Kernel command line: qemu=1 console=null android.checkjni=1
android.ril=ttyS0
<3>Unknown boot option `android.checkjni=1': ignoring
<3>Unknown boot option `android.ril=ttyS0': ignoring
<4>PID hash table entries: 512 (order: 9, 2048 bytes)
<4>Console: colour dummy device 80x30
<6>Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
<6>Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
<6>Memory: 96MB = 96MB total
<5>Memory: 94200KB available (2260K code, 461K data, 96K init)
<7>Calibrating delay loop... 321.12 BogoMIPS (lpj=1605632)
<4>Mount-cache hash table entries: 512
<6>CPU: Testing write buffer coherency: ok
<4>android_power_init
<4>android_power_init done
```



```
<4>yaffs Jan 22 2008 19:24:40 Installing.
<6>io scheduler noop registered
<6>io scheduler anticipatory registered (default)
<6>io scheduler deadline registered
<6>io scheduler cfq registered
<4>allocating frame buffer 320 * 480, got ffc10000
<4>Console: switching to colour frame buffer device 40x30
```



```
<6>binder: 669 close vm area 424a8000-428a8000 (4096 K) vma 2020071 pagep
5f
<6>binder_release: 669 threads 3, nodes 0 (ref 0), refs 2, active
transactions 0, buffers 0, pages 1
<6>binder_open: 695:695
<6>binder_mmap: 695 42641000-42a41000 (4096 K) vma 71 pagep 5f
<6>binder_flush: 695 woke 2 threads
<6>binder_open: 768:768
<6>binder_mmap: 768 42641000-42a41000 (4096 K) vma 71 pagep 5f
```

```
# cat /proc/meminfo
MemTotal: 94672 kB
MemFree: 17856 kB
Buffers: 0 kB
Cached: 46008 kB
SwapCached: 0 kB
Active: 44452 kB
Inactive: 25944 kB
SwapTotal: 0 kB
SwapFree: 0 kB
Dirty: 0 kB
Writeback: 0 kB
AnonPages: 24420 kB
Mapped: 18512 kB
Slab: 2412 kB
SReclaimable: 592 kB
SUnreclaim: 1820 kB
PageTables: 2004 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
CommitLimit: 47336 kB
Committed_AS: 337008 kB
VmallocTotal: 876544 kB
VmallocUsed: 32824 kB
VmallocChunk: 806908 kB
```

```
# cat /proc/mtd
dev: size erasesize name
mtd0: 04000000 00020000 "system"
mtd1: 04000000 00020000 "userdata"
mtd2: 04000000 00020000 "cache"
```

```
[tonyt@zonk ~]$ ls /usr/java/android-sdk_m5-rc14_linux-x86/tools/lib/images/
totale 54288
drwxrwx--- 3 tonyt tonyt 103 feb 12 02:11 ./
drwxrwx--- 3 tonyt tonyt 4096 feb 12 02:11 ../
-rwxrwxr-x 1 tonyt tonyt 1242736 feb 12 02:11 kernel-qemu*
-rw-rw-r-- 1 tonyt tonyt 690407 feb 12 02:11 NOTICE
-rw-rw-r-- 1 tonyt tonyt 311322 feb 12 02:11 ramdisk.img
drwxrwx--- 7 tonyt tonyt 69 feb 12 02:11 skins/
-rw-rw---- 1 tonyt tonyt 48812544 feb 12 02:11 system.img
-rw-rw---- 1 tonyt tonyt 4515456 feb 12 02:11 userdata.img
```



## Android on steroids (3)

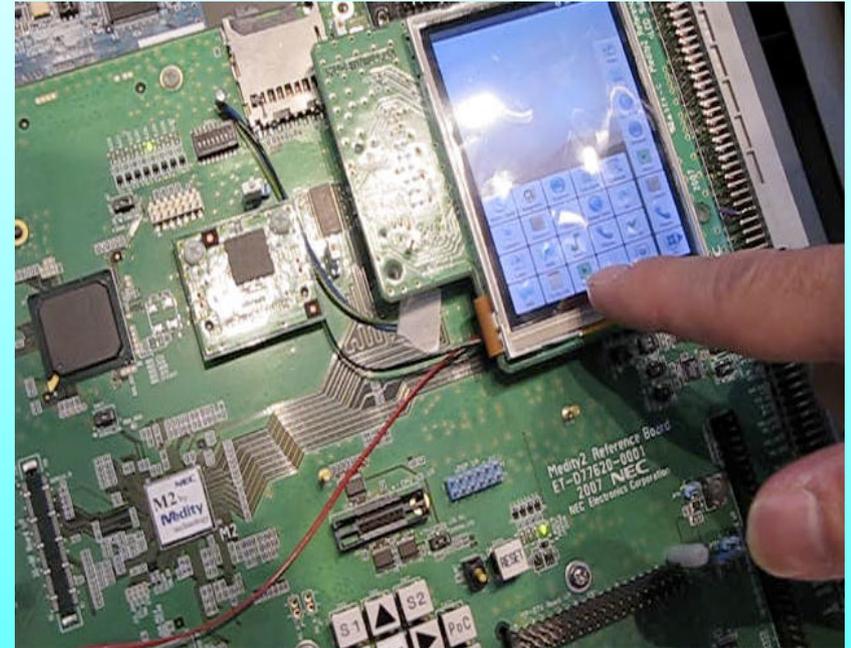
- Confronto con un kernel 2.6.23 *vanilla*:
  - arch/arm/mach-goldfish: supporto per QEMU
  - fs/yaffs2: molto veloce, solo per flash NAND
  - drivers/android: debug, gestione energia
  - drivers/misc/lowmemorykiller: basta la parola!
  - drivers/binder: per IPC, più sofisticato di COM
- OpenBinder usa `ioctl()` SU `/dev/binder`
- **Dianne Hackborn** ora lavora per Google, dopo le esperienze di Be e PalmSource



# Android: hardware vero

- La star del CES 2008...

**Prototipo NEC**



- ...ma anche la gioia di vari sviluppatori



**Atmark-Techno Armadillo-500**  
(Freescale i.MX31L)



# I processi in esecuzione (1)

- Poco diverso da un “normale” Linux

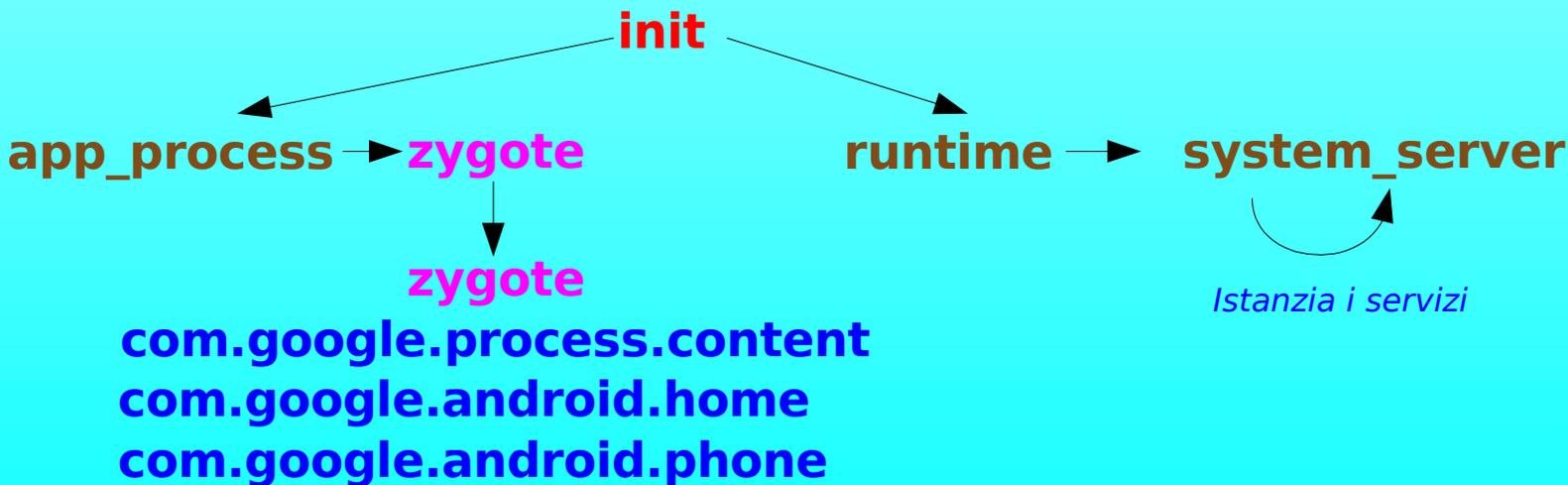
```
# ps
USER      PID    PPID    VSIZE  RSS      WCHAN    PC        NAME
root      1      0       248    160     c0084edc 0000ae2c S /init
root      2      0       0      0      c0049168 00000000 S kthreadd
root      3      2       0      0      c003ad20 00000000 S ksoftirqd/0
root      4      2       0      0      c00460ac 00000000 S events/0
root      5      2       0      0      c00460ac 00000000 S khelper
root      8      2       0      0      c00460ac 00000000 S suspend/0
root      33     2       0      0      c00460ac 00000000 S kblockd/0
root      36     2       0      0      c00460ac 00000000 S cqueue/0
root      38     2       0      0      c0153ab8 00000000 S kseriod
root      76     2       0      0      c005eb48 00000000 S pdflush
root      77     2       0      0      c005eb48 00000000 S pdflush
root      78     2       0      0      c00624f8 00000000 S kswapd0
root      79     2       0      0      c00460ac 00000000 S aio/0
root      201    2       0      0      c0151168 00000000 S mtdblockd
root      220    2       0      0      c00460ac 00000000 S kmmcd
root      233    2       0      0      c00460ac 00000000 S rpciod/0
root      492    1       5516   300     ffffffff 0000e0e4 S /sbin/adbd
root      493    1       2824   300     ffffffff afe0c79c S /system/bin/usbq
root      494    1       644    228     c0179838 afe0ca9c S /system/bin/debuggerd
root      495    1       12564  604     ffffffff afe0c79c S /system/bin/rild
root      496    1       55540  13952  c0179838 afe0ca9c S zygote
root      497    1       20248  2164   ffffffff afe0clfc S /system/bin/runtime
bluetooth 498    1       1224   776     c0084edc afe0d07c S /system/bin/dbus-daemon
root      513    1       134036 19204  ffffffff afe0clfc S system_server
root      541    496     55544  7712   c01d9aac afe0cbfc S zygote
app_0     542    541     71468  14616  ffffffff afe0d204 S com.google.process.content
root      565    492     732    308     c00386cc afe0ceac S /system/bin/sh
root      566    565     732    316     c00386cc afe0ceac S sh
app_4     599    541     69740  12576  ffffffff afe0d204 S com.google.android.home
phone     602    541     72904  14952  ffffffff afe0d204 S com.google.android.phone
app_5     695    541     89424  14700  ffffffff afe0d204 S com.google.android.samples
app_0     768    541     87020  19500  ffffffff afe0d204 S com.google.android.maps
root      854    566     840    324     00000000 afe0bfbc R ps
```



# I processi in esecuzione (2)

- **runtime**: dispatcher degli URI
- **app\_process**: lanciatore applicazioni; in particolare lancia **zygote**
- **system\_server**: controllore dei servizi, simile a svchost.exe

AudioFlinger  
 SurfaceFlinger  
 USB  
 activity  
 activity.providers  
 activity.receivers  
 activity.services  
 alarm  
 battery  
 checkin  
 content  
 hardware  
 isms  
 location  
 meminfo  
 notification  
 org.bluez.bluetooth  
 org.bluez.bluetooth\_service  
 package  
 phone  
 power  
 search  
 simphonebook  
 statusbar  
 volume  
 window





# I processi in esecuzione (3)

- La VM *Java-like* condivide il bytecode:

```
app_4 580 561 87344 15396 ffffffff afe0d204 S com.google.android.home

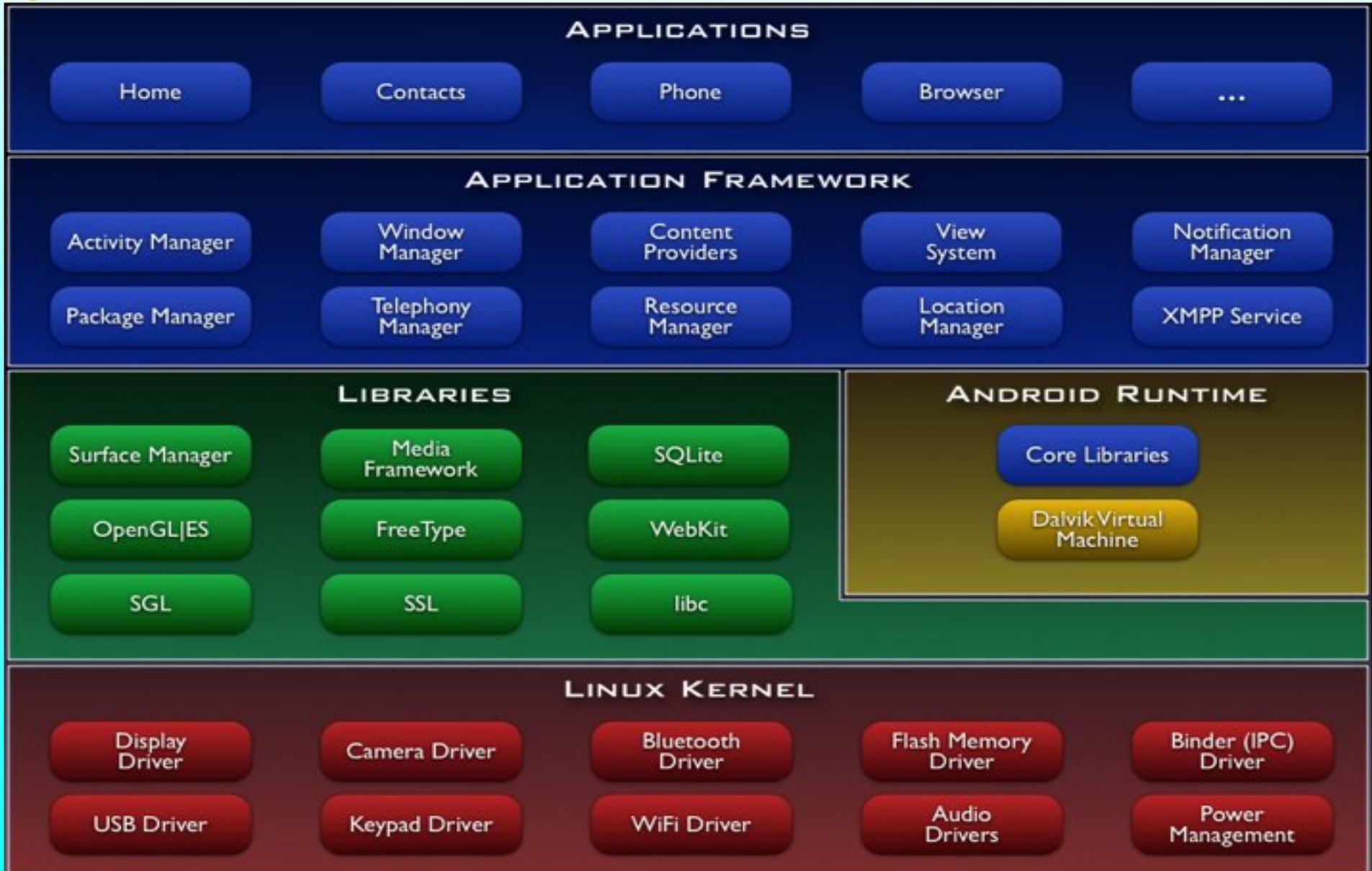
# showmap -v 580
virtual shared shared private private
size refer'd clean dirty clean dirty object
-----
 84      20      0      48      0      20 [stack]
 16       0      0      12      0       0 [lib_bss]
 76      44      44      4      0       0 /system/bin/linker
 44      28      0      0      0      28 [lib_bss]
 12       4      0      8      0       4 /system/lib/libc.so
228     156     152      4      0       0 /system/lib/libc.so

✂-
1024      4      0      0      0       4 /tmp/android-shared_heap.580.0x111ee8
(deleted)
148      96      96      0      0       0 /system/fonts/DroidSans.ttf
1024      4      0      0      0       4 /tmp/android-shared_heap.580.0x1515d0
(deleted)
220     140     140      0      0       0 /system/app/Maps.apk
 8        8      8      0      0       0 /system/app/Browser.apk

✂-
 36       0      0      0      0       0 /system/framework/ext.jar
5256    1344    1344      0      0       0 /data/dalvik-
cache/system@framework@core.jar@classes.dex
1884      0      0      0      0       0 /system/framework/core.jar
1020     908      0     884      0     136 [anon]
 8        4      0      0      0       4 [anon]
1540    1436      0      56      0    1436 /tmp/dalvik-heap-496 (deleted)
 32       4      4      0      0       0 /system_properties (deleted)
1020      4      0      0      0       4 [stack]
 4        0      0      0      0       0 /dev/zero (deleted)
1484    1040      0     624      0     860 [heap]
 8        8      4      4      0       0 /system/bin/app_process
-----
65308   13956   9148   2628   1040   2580 TOTAL
```



# Architettura di sistema





## Cosa è Dalvik? (1)

- Un villaggio dell'Islanda (**Dan Bornstein**)
- Il modo *preferito* per sviluppare le applicazioni Android è tramite Java
- JRE ritagliato da Java SE (Harmony Classlib) + package con licenza Apache
- Google non afferma che Dalvik sia una Java VM, aggira elegantemente la licenza
- Dalvik è solo interprete del bytecode
- Più efficiente di Sun JVM solo interprete



## Cosa è Dalvik? (2)

- Il risultato della compilazione non è bytecode in standard JVM (.dex)
- Formato ottimizzato per la condivisione
- Una istanza di Dalvik VM per processo
- Basata su registri anziché su stack?!

```
VFY: register1 v%u type %d, wanted %d
VFY: register2 v%u out of range (%d)
VFY: register2 v%u-%u values %d,%d
VFY: register1 v%u holds uninitialized ref
VFY: register2 v%u type %d, wanted %d
VFY: register1 v%u type %d, wanted ref
VFY: failed to get this from register %u
VFY: tried to get class from non-ref register v%d (type=%d)
VFY: register %u holds uninitialized reference
```

Stringhe  
estratte da  
[libdvm.so](http://libdvm.so)



## Cosa è Dalvik? (3)

- `dalvikvm/libdvm.so` compilate per Thumb!
- Però nessuna traccia di istruzioni Jazelle
- E' possibile scrivere applicazioni native, ma non sarebbero *sicure*

Un cellulare è  
un computer!



**Demo di Quake**  
[www.youtube.com/AndroidDevelopers](http://www.youtube.com/AndroidDevelopers)





# I package Java disponibili

android	java.io	junit.extensions
android.app	java.lang	junit.framework
android.awt	java.lang.annotation	org.apache.http
android.content	java.lang.instrument	org.apache.http.auth
android.content.pm	java.lang.ref	org.apache.http.auth.params
android.database	java.lang.reflect	org.apache.http.client
android.database.sqlite	java.math	org.apache.http.client.methods
android.drm	java.net	org.apache.http.client.methods.multipart
android.graphics	java.nio	org.apache.http.client.params
android.graphics.drawable	java.nio.channels	org.apache.http.client.protocol
android.graphics.drawable.shapes	java.nio.channels.spi	org.apache.http.client.utils
android.graphics.glutils	java.nio.charset	org.apache.http.conn
android.hardware	java.nio.charset.spi	org.apache.http.conn.params
android.location	java.security	org.apache.http.conn.ssl
android.media	java.security.acl	org.apache.http.conn.util
android.net	java.security.cert	org.apache.http.cookie
android.opengl	java.security.interfaces	org.apache.http.cookie.params
android.os	java.security.spec	org.apache.http.entity
android.provider	java.sql	org.apache.http.impl
android.sax	java.text	org.apache.http.impl.auth
android.telephony	java.util	org.apache.http.impl.client
android.telephony.gsm	java.util.concurrent	org.apache.http.impl.conn
android.telephony.gsm.stk	java.util.concurrent.atomic	org.apache.http.impl.conn.tsccm
android.test.mock	java.util.concurrent.locks	org.apache.http.impl.cookie
android.text	java.util.jar	org.apache.http.impl.entity
android.text.method	java.util.logging	org.apache.http.impl.io
android.text.style	java.util.prefs	org.apache.http.io
android.text.util	java.util.regex	org.apache.http.message
android.util	java.util.zip	org.apache.http.params
android.view	javax.crypto	org.apache.http.protocol
android.view.animation	javax.crypto.interfaces	org.apache.http.util
android.webkit	javax.crypto.spec	org.apache.http.util.concurrent
android.widget	javax.microedition.khronos.opengles	org.bluez
com.google.android.gtalkservice	javax.net	org.json
com.google.android.maps	javax.net.ssl	org.w3c.dom
	javax.security.auth	org.xml.sax.ext
	javax.security.auth.callback	org.xml.sax.helpers
	javax.security.auth.login	
	javax.security.auth.x500	
	javax.security.cert	
	javax.sound.midi	
	javax.sound.midi.spi	
	javax.sound.sampled	
	javax.sound.sampled.spi	
	javax.sql	
	javax.xml.parsers	



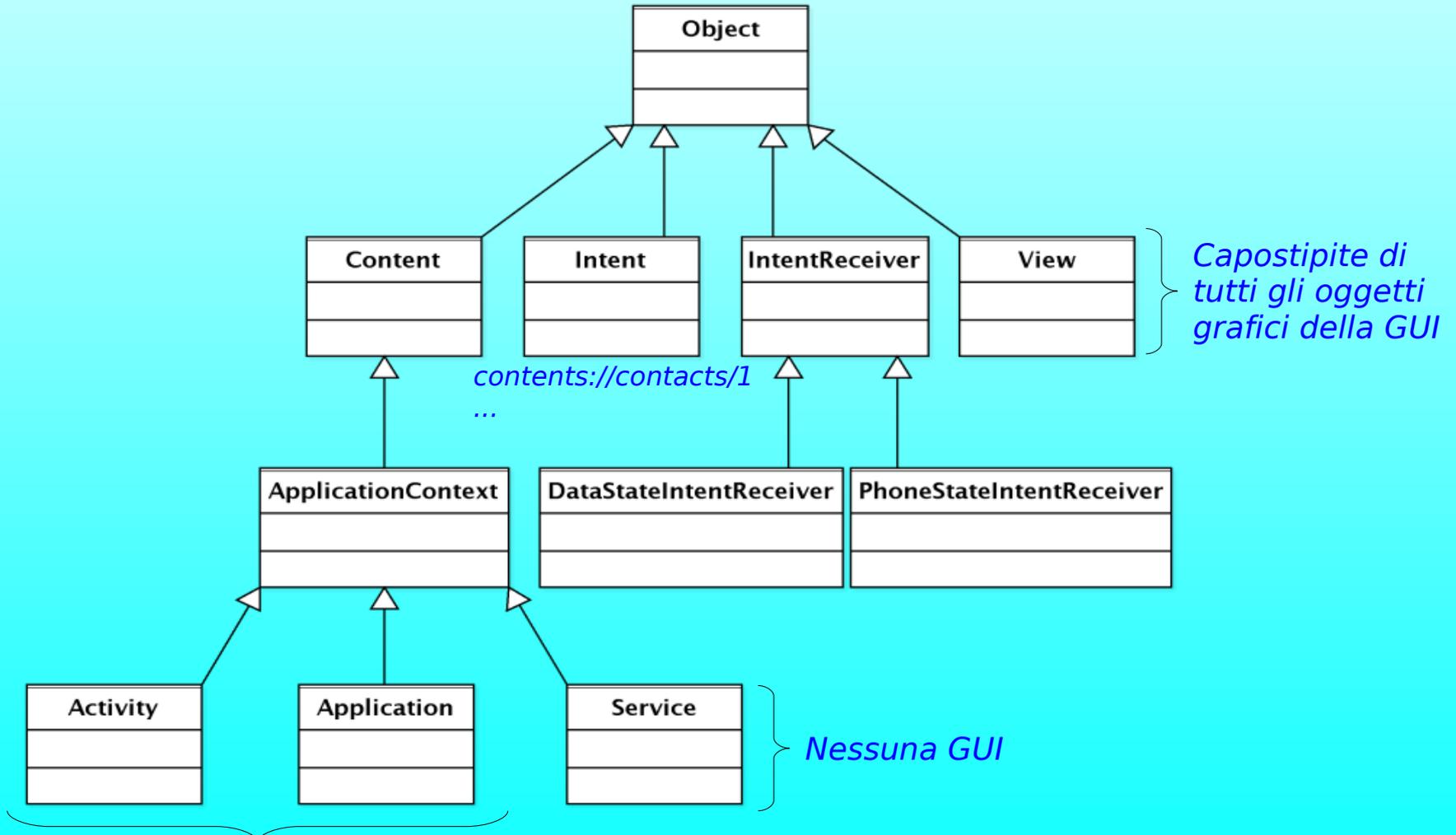
# Androidologia

- **Activity**: singola schermata d'applicazione
- **Intent**: per spostarsi fra le schermate; i dati sono espressi come URI
- **IntentReceiver**: listener Intent, GUI opzionale
- **Service**: in background, senza GUI
- **Content Provider**: consente di condividere dati con altre applicazioni
- **AndroidManifest.xml**: dichiara i componenti di un'applicazione e i loro prerequisiti



# Il modello degli oggetti

- Di solito, un'applicazione è una Activity



*Hanno una GUI* (CC by-nc-sa) Antonio Tringali 2008



# Configurazione applicazioni

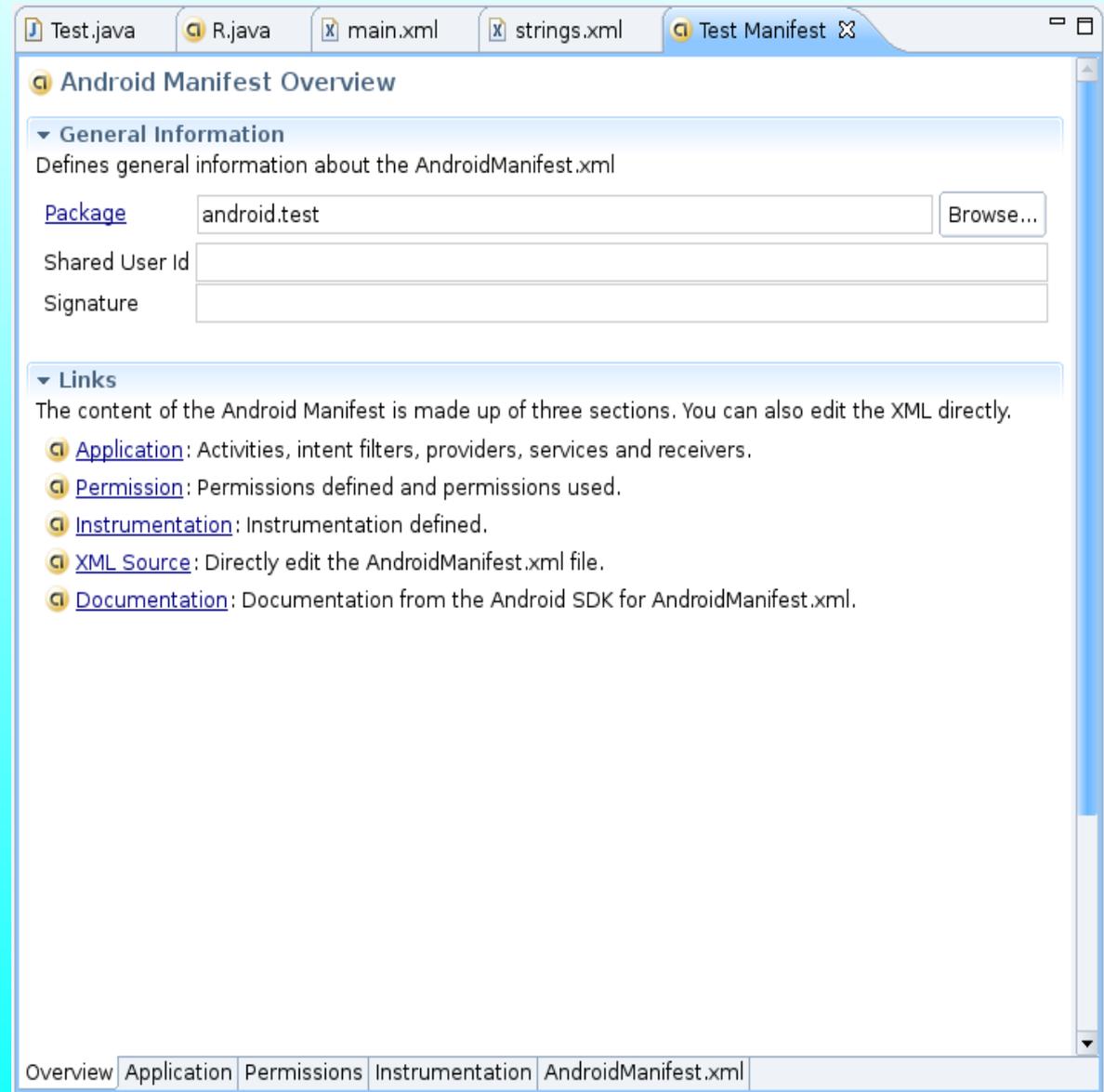
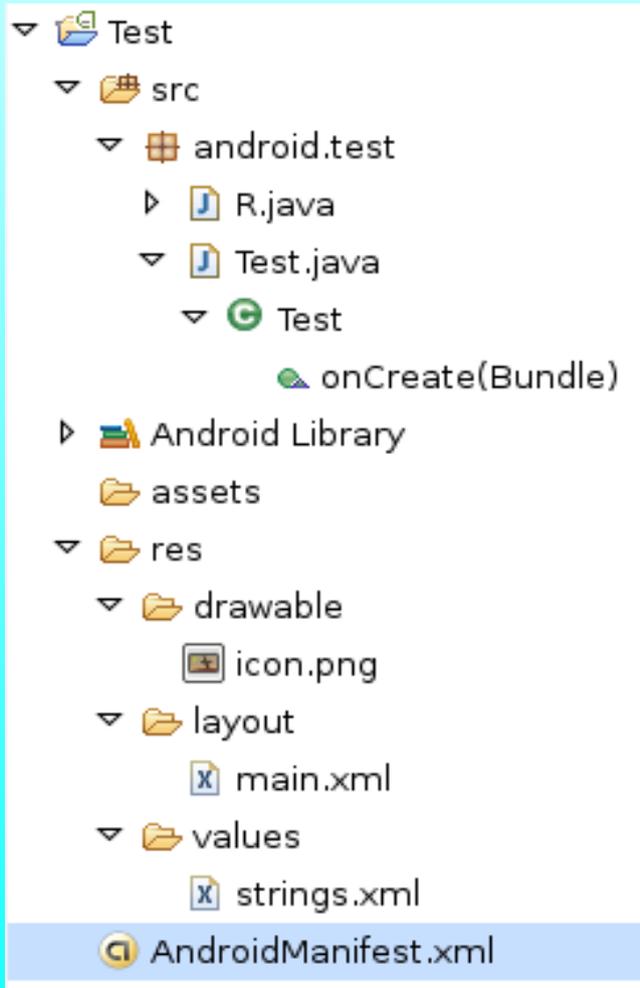
- In DB SQLite, molto Google-oriented ;-)
- Le config. di lancio in file XML
- Il resto sono file di config. Linux

```
# sqlite3
/data/data/com.google.android.providers.settings/databases/settings.db
SQLite version 3.5.0
Enter ".help" for instructions
sqlite> .tables
bluetooth_devices favorites gservices
bookmarks googlelogin system
sqlite> select * from favorites;
1|Dialer|
#action(android.intent.action.MAIN)launchFlags(4)component(com.google.
android.phone!com.google.android.phone.DialerActivity)|0
2|Contacts|
#action(android.intent.action.MAIN)launchFlags(4)component(com.google.
android.contacts!com.google.android.contacts.ContactsActivity)|1
sqlite> select * from gservices;
1|gservices-change-count|1
2|maps_server_url|http://www.google.com/glm/mmap/a
3|gtalk_hostname|talk.google.com
4|gtalk_port|5222
5|gtalk_secure_port|5223
6|gtalk_auth_sasl|false
7|gtalk_ssl|true
8|gtalk_compress|false
9|google_login_client_auth_host|www.google.com
10|google_login_client_default_auth_path|/accounts/ClientLogin
11|google_login_client_youtube_auth_path|/youtube/accounts/ClientLogin
12|google_login_connection_timeout|30000
13|google_login_generic_auth_service|mail
14|gmail_host|gmail.com
sqlite> select * from system;
1|music_volume|3
2|voice_volume|3
3|ringer_volume|3
4|dim_screen|0
5|stay_on_while_plugged_in|1
6|screen_off_timeout|30000
7|font_scale|1.0
sqlite> .exit
#
```



# Anatomia di un'applicazione (1)

- Layout delle dir





## Anatomia di un'applicazione (2)

- Il layout dei componenti e le label

```
Test.java R.java strings.xml main.xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <EditText
        android:id="@+id/edit_text"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/body_text"
        android:editable="true"
        android:autoLink="all"
        android:autoText="true"
    />
    <Button
        android:id="@+id/done_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/done_button_text" />
</LinearLayout>
```

```
Test.java R.java strings.xml main.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Io, Android</string>
    <string name="body_text">Scrivi del testo qui...</string>
    <string name="done_button_text">Fatto!</string>
</resources>
```



# Anatomia di un'applicazione (3)

- Per gestire il bottone sotto EditText:

```
Test.java R.java strings.xml main.xml
package android.test;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class Test extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        // Codice aggiunto da me
        Button doneButton = (Button) findViewById(R.id.done_button);
        doneButton.setOnClickListener(new Button.OnClickListener() {
            public void onClick(View v) {
                // Perform action on click
                EditText edit = (EditText) findViewById(R.id.edit_text);
                edit.setText(edit.getText(), TextView.BufferType.SPANNABLE);
            }
        });
        // Fine dell'aggiunta
    }
}
```



# Anatomia di un'applicazione (4)

- E un po' di residui di lavorazione...

The image shows two side-by-side windows from an IDE. The left window displays the content of `R.java`, which is an auto-generated file containing resource IDs. The right window shows the `AndroidManifest.xml` editor, which is currently in the 'Application' tab. This tab allows for configuring application-level components and attributes.

```
/* AUTO-GENERATED FILE. DO NOT MODIFY.
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found. It
 * should not be modified by hand.
 */

package android.test;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class id {
        public static final int done_button=0x7f050001;
        public static final int edit_text=0x7f050000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040000;
        public static final int body_text=0x7f040001;
        public static final int done_button_text=0x7f040002;
    }
}
```

**Android Manifest Application**

**Application**  
The application tag describes application-level components contained in the package, as well as general application attributes.  
[Documentation](#): Documentation for the application tag in the AndroidManifest.xml.  
 Define an <application> tag in the AndroidManifest.xml

**Application Attributes**  
Defines the attributes specific to the application.

Class Name	<input type="text"/>	Browse...	Process	<input type="text"/>
Label	<input type="text"/>	Browse...	Task Affinity	<input type="text"/>
Icon	<input type="text" value="@drawable/icon"/>	Browse...	Theme	<input type="text"/> Browse...
Persistent	<input type="text" value="&lt;undefined&gt;"/>			

**Application Nodes**  
List of all elements in the application

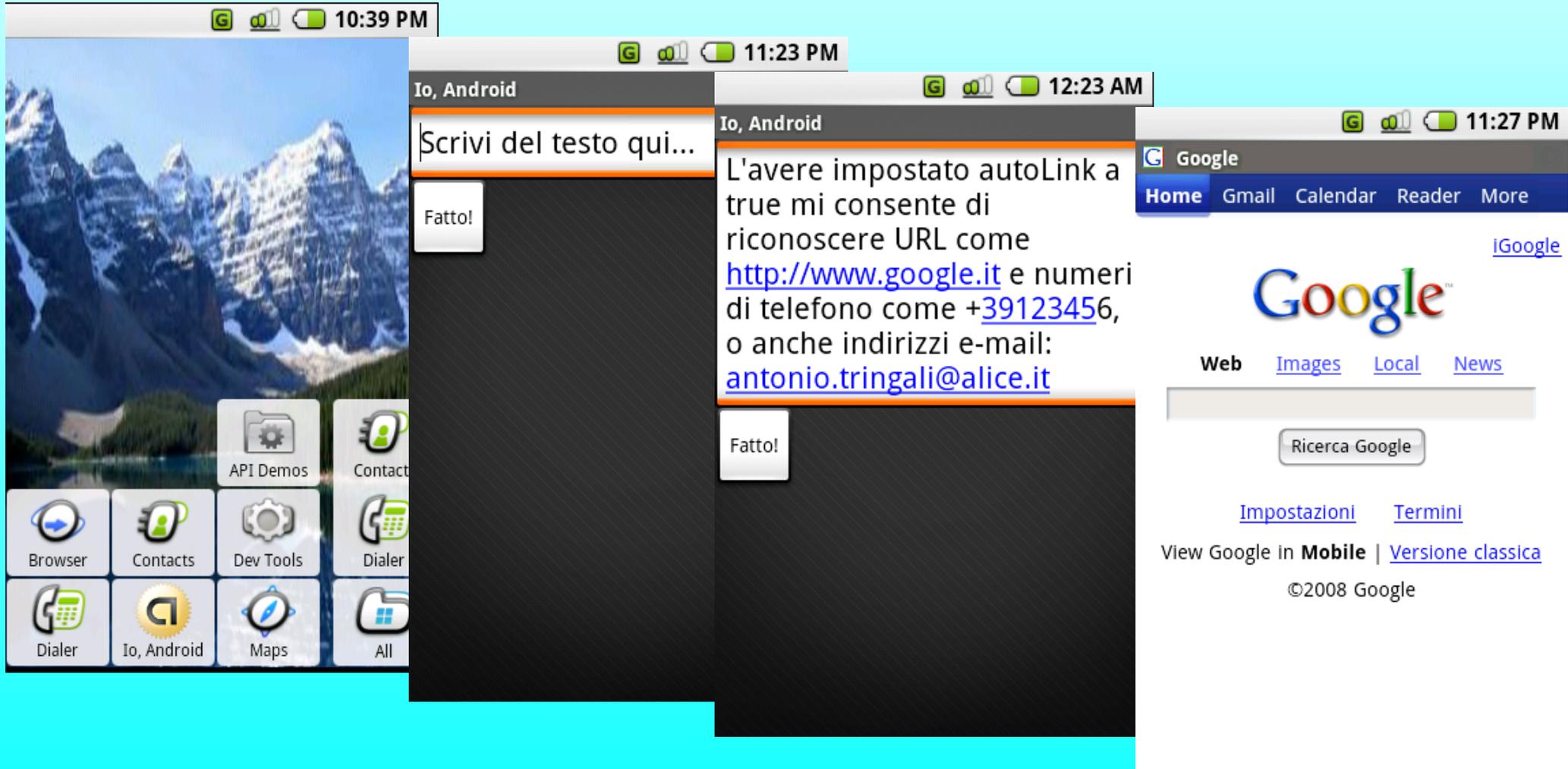
- ▼ **.Test (Activity)**
- ▼ Intent Filter
- A android.intent.action.MAIN
- C android.intent.category.LAUNCHER

Overview Application Permissions Instrumentation AndroidManifest.xml



# Anatomia di un'applicazione (5)

- E finalmente il risultato di tanta (!) fatica





# Applicazioni presenti

- A corredo con l'emulatore:

```
# ls -l /system/app
-rw-r--r-- root    root          6899 2008-02-12 02:10 GoogleAppsProvider.apk
-rw-r--r-- root    root         31137 2008-02-12 02:10 MasfProxyService.apk
-rw-r--r-- root    root         19909 2008-02-12 02:10 MediaProvider.apk
-rw-r--r-- root    root         13658 2008-02-12 02:10 SettingsProvider.apk
-rw-r--r-- root    root         24016 2008-02-12 02:10 TelephonyProvider.apk
-rw-r--r-- root    root         22494 2008-02-12 02:09 ContactsProvider.apk
-rw-r--r-- root    root        477842 2008-02-12 02:11 Browser.apk
-rw-r--r-- root    root        115750 2008-02-12 02:11 Contacts.apk
-rw-r--r-- root    root         99623 2008-02-12 02:11 Development.apk
-rw-r--r-- root    root          8542 2008-02-12 02:11 Fallback.apk
-rw-r--r-- root    root          6555 2008-02-12 02:11 GTalkSettings.apk
-rw-r--r-- root    root       230116 2008-02-12 02:11 Home.apk
-rw-r--r-- root    root         44891 2008-02-12 02:11 GoogleApps.apk
-rw-r--r-- root    root       221523 2008-02-12 02:11 Maps.apk
-rw-r--r-- root    root       448665 2008-02-12 02:11 Phone.apk
-rw-r--r-- root    root         19390 2008-02-12 02:11 ImProvider.apk
-rw-r--r-- root    root       175694 2008-02-12 02:11 gtalkservice.apk
```



## ...e previste

- Lanciando manualmente

/system/bin/runtime:

```
# ls -l /system/app
-rw-r--r-- root root 6899 2008-02-12 02:10 GoogleAppsProvider.apk
-rw-r--r-- root root 31137 2008-02-12 02:10 MasfProxyService.apk
-rw-r--r-- root root 19909 2008-02-12 02:10 MediaProvider.apk
-rw-r--r-- root root 13658 2008-02-12 02:10 SettingsProvider.apk
-rw-r--r-- root root 24016 2008-02-12 02:10 TelephonyProvider.apk
-rw-r--r-- root root 22494 2008-02-12 02:09 ContactsProvider.apk
-rw-r--r-- root root 477842 2008-02-12 02:11 Browser.apk
-rw-r--r-- root root 115750 2008-02-12 02:11 Contacts.apk
-rw-r--r-- root root 99623 2008-02-12 02:11 Development.apk
-rw-r--r-- root root 8542 2008-02-12 02:11 Fallback.apk
-rw-r--r-- root root 6555 2008-02-12 02:11 GTalkSettings.apk
-rw-r--r-- root root 230116 2008-02-12 02:11 Home.apk
-rw-r--r-- root root 44891 2008-02-12 02:11 GoogleApps.apk
-rw-r--r-- root root 221523 2008-02-12 02:11 Maps.apk
-rw-r--r-- root root 448665 2008-02-12 02:11 Phone.apk
-rw-r--r-- root root 19390 2008-02-12 02:11 ImProvider.apk
-rw-r--r-- root root 175694 2008-02-12 02:11 gtalkservice.apk
```

- ...e rimosso (per il momento)  
android.speech.recognition!

AlarmClock.apk  
AlarmProvider.apk  
Anagrams.apk  
ApiDemos.apk  
Bounce.apk  
Calculator.apk  
Calendar.apk  
CalendarProvider.apk  
Camera.apk  
DataSets.apk  
Drm.apk  
GDataFeedsProvider.apk  
GTalkDiagnostics.apk  
GlobalTime.apk  
Gmail.apk  
GmailProvider.apk  
IM.apk  
LocalePicker.apk  
Mms.apk  
Music.apk  
NotePad.apk  
PackageInstaller.apk  
Settings.apk  
SetupWizard.apk  
Snake.apk  
Street.apk  
Sync.apk  
SyncProvider.apk  
Talk.apk  
Term.apk  
ToDoList.apk  
Updater.apk  
Vending.apk  
VoiceDialer.apk  
Voicemail.apk  
YouTube.apk



## Developers ^3



- Per stimolare l'estro degli sviluppatori Google ha lanciato un concorso da \$10M
- Sono esclusi i paesi canaglia: Iran, ...
- ... e anche Quebec e Italia per via della burocrazia, anche se qualcuno ne dubita
- Nonostante gli “incentivi”, sviluppare per Android è semplice e divertente
- Operativi pochi minuti dopo aver effettuato l'installazione dell'SDK



## Il futuro prossimo è remoto

- I PDA hanno mostrato il 40% di declino nelle vendite tra il 2006 e 2007
- Nel 2008 si stima saranno venduti oltre *un miliardo* di nuovi cellulari
- Ci sono già oltre *tre miliardi* di cellulari
- In molti posti del mondo l'unico telefono che conoscono è in tecnologia cellulare
- Molti fruiranno Internet attraverso il telefono: cambio di paradigma



## Ritorno al mondo nuovo

- La 4G di telefonia cellulare è su IPv6
- E' probabile che gli operatori spingano un canone fisso per servizi all-IP, con A/V
- Finalmente la realizzazione del sogno di *"The Dawn of the Stupid Network"*
- Nuove forme di introiti: acquisto di canzoni *on-line* o in dotazione con il telefono, giochi, servizi innovativi (?)
- Una *connessione sociale* pervasiva



# A ovest niente di nuovo

- Motorola DynaTAC 8000x (Dr. Martin Cooper, 1983)



	<i>Google/OHA Android</i>	<i>FIC/OE OpenMoko</i>	<i>Nokia Maemo</i>	<i>Trolltech Qtopia</i>	<i>Motorola MotoDev</i>	<i>ACCESS Linux Platform</i>	<i>Apple iPhone</i>
<b>Licenze</b>	Apache v2, GPL2, LGPL	GPL2 e LGPL	GPL2, LGPL, Apache, ...	Duale GPL2 e proprietaria	GPL2 kernel, il resto proprietario	GPL, ..., NetFront proprietario	Apple
<b>SDK</b>	Completo + Eclipse plug-in	Vari	VistaMax o Eclipse plug-in	Vari	Motorola + Eclipse plug-in	Vari	Xcode
<b>Linguaggio primario</b>	Java, con C/C++ possibili	Linguaggi con binding GTK+	Linguaggi con binding GTK+	Qt C++	Java ME, Qt C++	Vari	Objective-C
<b>Emulatore</b>	Android (QEMU)	QEMU	QEMU	QEMU	MotoDev e Java ME emulator	QEMU	Aspen
<b>Dispositivi</b>	N/A	Neo1973	Nokia N Series	Decine, Neo1973	Vari telefoni Motorola	Centinaia	iPhone, iPod Touch
<b>Costo Servizi</b>	N/A	Gratuiti	Gratuiti	Gratuiti	Alcuni servizi a pagamento	A pagamento	A pagamento (iTunes)



# Riferimenti

- <http://code.google.com/android/>  
Il sito principale di Android: download SDK, documentazione e forum
- <http://www.youtube.com/AndroidDevelopers>  
Video sull'architettura e le possibilità di Android (*Androidology Parti 1, 2, 3*)
- <http://www.dalvikvm.com>  
Reverse-engineering del formato dei file .dex
- <http://www.openhandsetalliance.com>  
Il sito ufficiale della Open Handset Alliance
- <http://www.anddev.org>  
"Scoperte" su Android e numerosi tutorial
- <http://www.linuxdevices.com/news/NS4262102607.html>  
Esempi di installazione di Android su vario hardware
- <http://isen.com/papers/Dawnstupid.html>  
*"The Dawn of the Stupid Network"*, David S. Isenberg, ACM Networker 2.1, February/March 1998





IO, ANDROID

# Domande?

