The Penguin and the Droid

Robert Berger

Reliable Embedded Systems

© 2012 by Robert Berger - Reliable Embedded Systems
CC-Licence: http://creativecommons.org/licenses/by-nc-sa/3.0/

last commit: 2012-06-15 18:48:02 +0300
revision: d2e157ef93b6b5e135a3a00cc975f50f2eaec79d

June 24, 2012
The views, opinions, positions or strategies expressed by the author and those providing comments are theirs alone, and do not necessarily reflect the views, opinions, positions or strategies of anybody else.
Let me introduce myself

Robert Berger
Embedded Software Specialist
(+30) 697 593 3428

http://www.ReliableEmbeddedSystems.com
Robert.Berger@ReliableEmbeddedSystems.com

Building World Class World Wide Win/Win Cooperations By Helping You To Create Better Embedded Software!
Roles

father
husband
consultant
Embedded SW

sharpen the saw

owner
Embedded Software Specialist

lecturer
Region 8
PA

Reliable Embedded Systems
The Penguin and the Droid

Figure: Tux: Linus Torvalds™?

Figure: Droid: George Lukas™
The Penguin and the Droid

Figure: The Penguin (Tux)

Figure: The Droid (Android Logo)
Why stay here?

• **Audience**
  • You work for a small/medium sized company which thinks about switching to Android
  • You don’t but would still like to hear my view of things (it will be fun)

• **Objective**
  • I would like to make you think out of the box and forget about the marketing hype for a while
  • ... and yes I’ll try to be provocative ...

• Please throw only soft items at me if you disagree with me
Linux is, in simplest terms, an operating system. Linux distribution’s makers have decided which kernel, operating system tools, environments, and applications to include and ship to users. There are, at last count, over 350 distinct distributions of Linux.

http://www.linuxfoundation.org/what-is-linux,
http://distrowatch.com/

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

Linux history

Figure: Memorable Linux Milestones
Figure: The Android Story
Linux vs. Android Kernel

Figure: Linux vs. Android Kernel releases
Linux openness 1

development process
  • created, maintained, improved by the community
release cycle
  • 1/2 year Ubuntu
  • 2 months kernel
  • all in the open
Linux openness 2

license

• user space: various (typically open source)
• kernel: GPLv2 (Copyleft not Copyright) guarantees the following freedoms (as in free speech, not free beer)
  • run the program as you wish for any purpose
  • study the source code and change it to do what you want
  • make copies and distribute them
  • distribute modified versions
  • requires you to make the modified source code available

Bazaar model
development process
- created behind Google’s (and partners) closed doors
- we don’t know how it’s maintained and improved

release cycle
- no fixed release cycle of the Android SDK
  - as a partner of Google (NDA) you get more frequent releases than the public
  - Motorola’s Xoom tablet was on the market quite a bit before a public release of Android 3.0 (Honeycomb) was made (Amazon Kindle, Facebook,...)
license

- Google did a lot to avoid the GPL and to be able to use a "permissive"/"less free" license
  - even replaced the standard glibc with bionic
  - like this proprietary modifications (without making source code available) are possible
  - for product differentiation
  - is this really necessary for "infrastructure code"?

- user space: Apache 2.0 licence wherever applicable
- kernel: GPLv2

Cathedral model
Linux Architecture

Figure: Linux architecture

user-space
- application(s)
  - use whatever framework/language you like

kernel
- system call interface
- monolithic
- process management, memory management, file systems, device drivers,...
Linux vs. Android Architecture

Figure: Linux architecture

Figure: Android architecture
user space

- Dalvic VM: for systems with limited memory, low power consumption, executes Dex bytecode
- JNI: access native code
- Zygote: pre-initializes VM instances and forks them
- Bionic instead of glibc
- additional Android stuff

Figure: Android architecture
Android Architecture

user space
- apps (default, yours, from market)
- application API (Java libs)
- on top of VM

... kernel
- see above

Figure: Android architecture
How much Linux is Android?

Bionic

- not glibc compatible
  - you’ll find glibc programs below /external folder which omit certain glibc features
- not POSIX compatible
  - lacks SysV or POSIX IPC (message queues, semaphores and shared memory) because when improperly used can cause resource leaks
  - supports UNIX and internet sockets and pipes
  - Binder replaces (more or less) message queues - IDL (Interface Definition Language)
  - ashmem replaces (more or less) shared memory
How much Linux is Android?

Android

• does not include a full set of standard Linux utilities
• has no native windowing system - (SurfaceFlinger)
• is not FHS (file hierarchy standard) compatible
• uses intents instead of DBUS
• is built on top of a Linux kernel
  • needs out of mainline patches for Android to work
  • opportunistic suspend not mainline compatible
  • Android mainlining project
How to get Android on your board?

Linux first?
• port Embedded Linux to hardware platform
• add Android extensions to it

Non-mainline Android?
• non-mainline kernel with Android extensions
• start from there

"Android capable" mainline Kernel?
• add Android user-space to it
Market for Embedded Linux

Figure: Hard Disc Recorder

Figure: Particle Accelerator

Figure: TV

Figure: Alarm Clock
Market for Android (Google)

Figure: Mobile Phone

Figure: Tablet

Figure: Tablet

Figure: Mobile Phone
Market for Android (not Google)

Figure: In car entertainment

Figure: In flight entertainment

Figure: Scope
Android Market share

Figure: Q4 2011, Nielsen Mobile Insights

250 Million Android Devices Activated, 11 Billion Apps Downloaded
Google philosophy: You can make money without doing evil.
http://www.google.com/about/company/tenthings.html

Google (still) wants Android only for the consumer market (mobile phones/tablets)

- to avoid fragmentation
- no other markets and even not everyone in the mobile phone/tablet market
- "Android Market is only licensed to handset manufacturers shipping devices. For questions about specific cases, contact android-partnerships.at.google.com." http://source.android.com/faqs.html
Google can make more money in mobile than it can on the desktop with the opportunities in the mobile search market.

Gomez-Barroso, Jose Luis; Feijoo, Claudio; Compano, Ramon; "Opportunities in the Mobile Search Market,"


pro

- high level of uniformity
- well partitioned managed and unmanaged code
- out-of-memory management, power management, logging, debugging,...
- well suited for devices with touch and graphical HMI
con

• kernel branch of Linux kernel
  • Android mainlining project - 3.3 kernel?
  • Android user space on top of Android-compatible mainline kernel?

• Google defines roadmap (consumer/mobile phone,...) - similar to working with non mainline Linux project
  • major effort to upgrade to newer versions
  • even for handset manufacturers - buy new phone

• Licenses
  • permissive
  • Microsoft gets license fees from 70% of all Android devices sold in U.S. [link](http://www.unbiasedtech.com/microsoft-lg-android-license-agreement/)
    • Google-Motorola deal
  • Can my device use apps from the Android market if I’m not a handset manufacturer?
### Android mainlining project status

<table>
<thead>
<tr>
<th>feature</th>
<th>kernel ver</th>
</tr>
</thead>
<tbody>
<tr>
<td>logger</td>
<td>staging 3.3 2012.03.19 partly 3.4, more in 3.5 partly 2.6.38 staging 3.3 2012.03.19</td>
</tr>
<tr>
<td>wakelocks</td>
<td>staging 3.3 2012.03.19</td>
</tr>
<tr>
<td>alarm timers</td>
<td>staging 3.3 2012.03.19 partly 2.6.38</td>
</tr>
<tr>
<td>ashmem</td>
<td>not mainlined?</td>
</tr>
<tr>
<td>network security</td>
<td>staging 3.3 2012.03.19 remove from 3.3-rc1 staging ?</td>
</tr>
<tr>
<td>binder</td>
<td>?</td>
</tr>
<tr>
<td>pmem driver</td>
<td>not needed?</td>
</tr>
<tr>
<td>USB gadget driver</td>
<td>?</td>
</tr>
<tr>
<td>timed gpio</td>
<td>?</td>
</tr>
<tr>
<td>timed output</td>
<td>?</td>
</tr>
<tr>
<td>low memory killer</td>
<td>?</td>
</tr>
<tr>
<td>ram console</td>
<td>?</td>
</tr>
<tr>
<td>ion graph.mem.drv</td>
<td>?</td>
</tr>
</tbody>
</table>
Thank you!

Robert Berger
Embedded Software Specialist
(+30) 697 593 3428

http://www.reliableembeddedsystems.com
Robert.Berger@ReliableEmbeddedSystems.com

Building world class world wide win/win cooperations by helping you to create better embedded software!