ReliaBleEmBeddedSystems

The Penguin and the Droid - A comparison

Robert Berger - Reliable Embedded Systems - Consulting Training Engineering
http://www.ReliableEmbeddedSystems.com
robert.berger@ReliableEmbeddedSystems.com

last commit: *2012-02-20 13:52:59 +0200*
revision: *eac8584b81ac72c6d2540f53a2e5f18a30a94eaf*

Disclaimer

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.

Acknowledgement

I would like to thank Chris Simmonds [1] and Manuel Di Cerbo [2] for their valuable input to this document.

Intro

What is Linux? [3]

This is how the Linuxfoundation describes it:

> 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. [4]

What is Android? [5]

According to the Android developers website [6]:

> Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK [7] provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

- More official Linux than Android distros - Linux more diverse.
- Linux's application development not limited to Java.
- The OS used by Android is a fork of the Linux kernel.

History

Linux

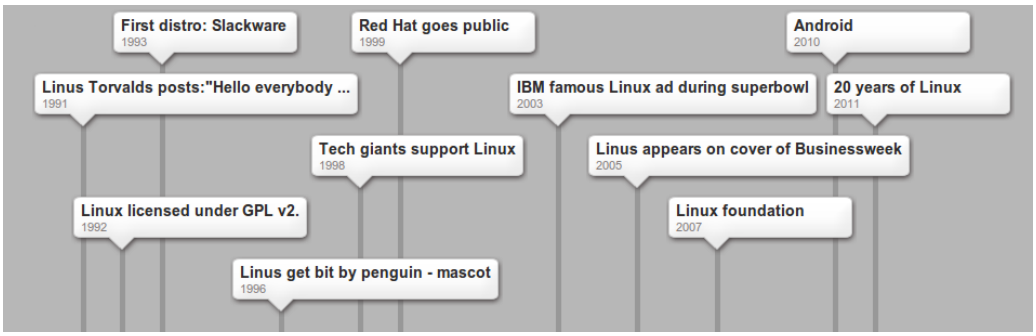

Figure 1: Memorable Linux Milestones [8] [9]

Nowadays Linux runs on many different kinds of devices. From wrist watches to supercomputers. Cool devices are TiVo, the large Hadron Collider, a fridge, a TV, Chumby, a navigation system, Kindle[1], a self driving car,... [10]

Android

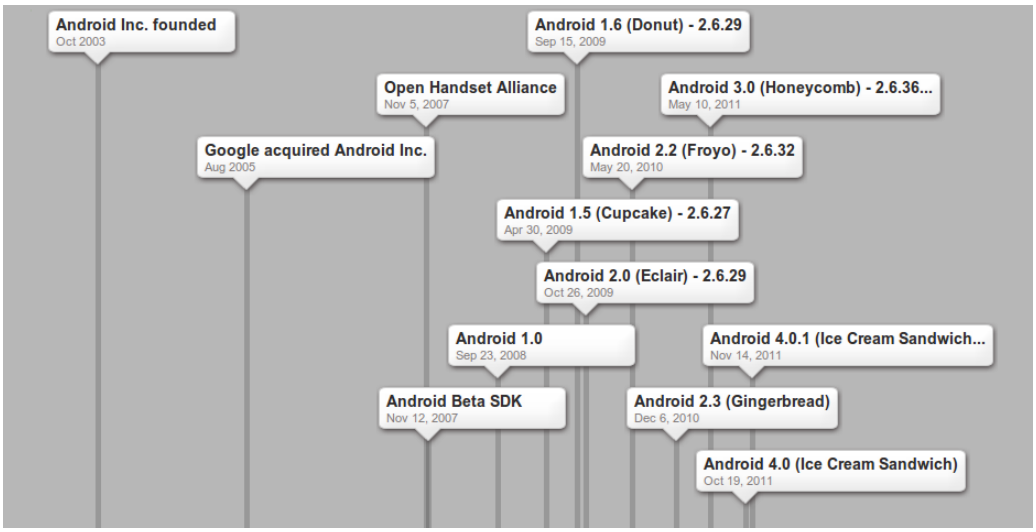

Figure 2: The Android Story [11] [12]

As of November 16, 2011, during the Google Music announcement "These Go to Eleven", 200 million Android devices had been activated. Based on this number, with 1.9% of Android devices being tablets, approximately 3.8 million Android Honeycomb Tablets have been sold. [13] According to Larry Page, Android has now (19th of Jan. 2012) been activated on over 250 million devices and the Android Marketplace has managed to cruise past its 11 billionth download. [14]

- There is a gap between the Linux kernel release and the Android (kernel) release - see figure 3.
- Android is critized for its fragmentation problem: many devices with different specs - hard to know what it takes to run the latest Android release on your device.

[1] In the meantime there is a hack available to run an Android on the Kindle Fire which can also download applications from Google's Market and not just from Amazon's Appstore. [15]

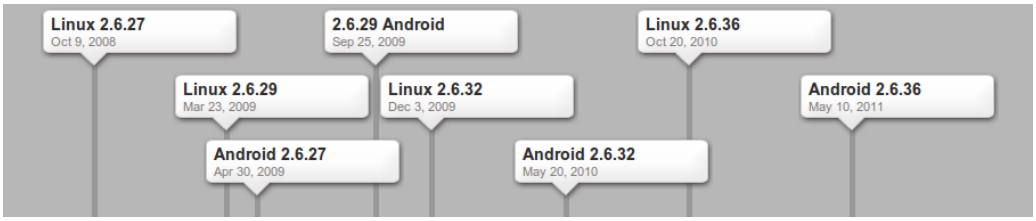

Figure 3: Linux vs. Android Kernel releases [9] [12] [16]

Openness

Linux

Linux distributions are released according to a schedule like e.g. the half year cycle of Ubuntu. [17] A new stable version of the Linux kernel is released roughly every two months. To simplify things let's concentrate on the development process on the Linux kernel. It's an open process, which is created, maintained and constantly improved by "the community". This is an approach which can be categorized as a "Bazaar" model, as described in the famous book by Eric S. Raymond called "The Cathedral and the Bazaar". [18] The Linux kernel is licenced under the GPL v2. [19] Various other packages in a typical Linux distribution may follow other licensing terms, but are usually covered by one of the many open source licenses. [20]

Android

The development process is defined behind closed doors and input comes only from Google and it's business partners. This resembles a "Cathedral" [18] in the spirit of Eric S. Raymond's book.

Google did a lot (even replaced the standard C library) to avoid the GPL and to be able to use a "permissive"/"less free" license. The reasoning here is that proprietary modifications for product differentiation would be allowed because of this.[2] The preferred license for the Android Open Source Project is the Apache Software License, 2.0 ("Apache 2.0"), and the majority of the Android software is licensed with Apache 2.0. While the project will strive to adhere to the preferred license, there may be exceptions which will be handled on a case-by-case basis. For example, the Linux kernel patches are under the GPLv2 license with system exceptions. [21]

There is no real fixed public release cycle for Android SDKs and some say on purpose. SDKs are released to business partners of Google under an NDA more frequently than to the public. An example is the delayed public release of Android 3.0 (Honeycomb) [22] [23] as already mentioned above in figure 2. Does this mean that although or because "Open Android" comes from the "Open Handset Alliance" [24] it's more "open" for some Google partners than for the public?

- The development process of Android(Cathedral) is defined behind closed doors, while for Linux it's open(Bazaar). One might argue that "Open Source" means sharing control with the community, not just making sources available for download from time to time. [25]
- Linux components are typically controlled by the community and hence open source. Android is controlled by Google and it's business partners and releases are done first to strategic partners and afterwards to the public.
- Motorola's Android 3.0 (Honeycomb)-fueled Xoom tablet has been on the market for some time without releasing the sources, which made Google's decision to hold the code from wide release a bit mystifying. [26]
- There are rumors that the delay of Honeycomb was caused by an attempt of Amazon to create an Android fork for it's Kindle tablet [27] and by the Facebook guys who seem to be working on an Android fork as well. [28]

[2]The author strongly believes that there is "infrastructure" code like e.g. the Linux kernel which is not differentiating the product and should be maintained by the community. The differentiating code is typically on higher software layers.

- Maybe the licensing decision was commercially a clever move after all, since Android might free itself from much of the patent and intellectual property FUD (fear and doubt) surrounding it. [29]
- Continued growth and market traction for Android will supersede legal battles that become less of a concern or priority for all parties in 2012. [29]

Architecture

Linux

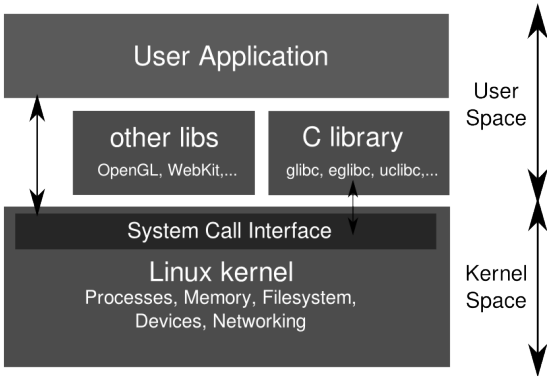

Figure 4: Linux Architecture

Figure 4 depicts the Linux System Architecture. A distinction is made between kernel- and user-space. The Linux kernel is ”monolithic” and contains facilities for process management, memory management, file systems, device drivers, networking facilities,... The system call interface is necessary for the communication between user- and kernel space and back. System calls are usually done by libraries like the C-library, but you can also call them directly form your user-space application, although this should be avoided. The top level shows your application. There are no explicit restrictions regarding your choice of frameworks, libraries and programming language(s). Pick whatever you are fancy!

Android

As you can see in figure 5 the general idea is pretty much like Linux plus a JVM (Java Virtual machine). There is a Linux kernel at the bottom with some extra Android stuff. Just on top of it are libraries (C/C++ libraries, Surface manager which handles UI Windows, 2D and 3D graphics, Media codecs, SQLite, Webkit...). Please note that the ”standard” C library has been replaced by `Bionic`. Above is the `Dalvic` virtual machine, which is optimized for limited memory and power consumption. It executes `Dex` bytecode, which is more compact than Java classes. Lower level code could be accessed through the JNI (Java Native Interface). `Zygote` pre-initializes VM instances and forks them. On top of the VM are Java libraries which expose an Application API to Android's default apps, your apps and apps from the market. The `Activity Manager` manages an application's life-cycle. Note that Android adds an extra component called `Binder` for IPC (inter process communication).

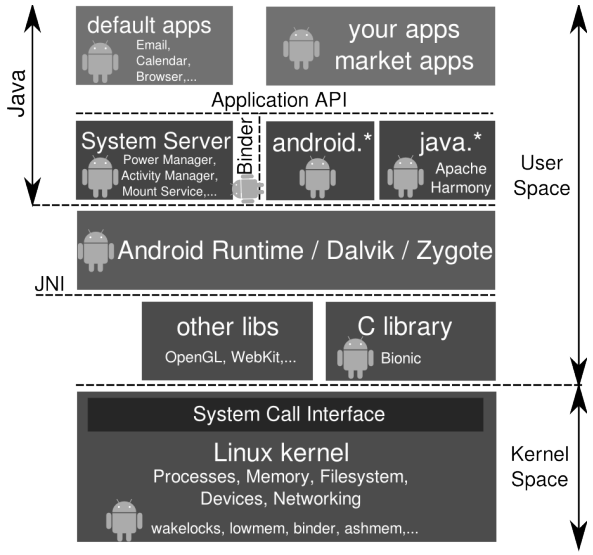

Figure 5: Android Architecture

How much Linux is Android?

Compared to Linux Android has a ”high level of uniformity” and well partitioned managed and unmanaged code. It is an attractive choice for applications which require touch and a graphical human machine interface.

- Android's `Bionic` is not entirely `glibc` compatible.[3]
- Android's `Bionic` is not entirely POSIX compatible since it lacks `SysV` or `POSIX` IPC (`message queues`, `semaphores` and `shared memory`)[4]
  - The bionic documentation[5] claims that `SysV` and `POSIX` IPC can cause resource leaks, which can be used by buggy or malicious applications to bring a system down.
  - `Binder`[6] replaces (more or less) `message queues`.
  - `ashmem` replaces (more or less) `shared memory`.
- Android does not include a full set of standard Linux utilities.
- Android has no native windowing system. It uses `SurfaceFlinger` instead of e.g. `Qt-Embedded` on Embedded Linux.[7]
- Android is not compatible with the FHS (Filesystem Hierarchy Standard) [30].
- Android uses `intents` instead of `DBUS` which is used in Linux.
- Android is built on top of a Linux kernel, but it currently needs out of mainline patches for Android to work.
  - This means that drivers written for Android might not work with mainline Linux and vice versa - e.g. drivers which use Android's `opportunistic suspend` are currently not compatible to mainline Linux.
  - The ”Android Mainlining Project” [31] is a community effort to mainline all patches required to run the current released version of Android.

Either Embedded Linux is first ported to your hardware platform and Android extensions are added to it, or you could get a non-mainline kernel with Android extensions and start from there. Hopefully sooner or later we'll have an ”Android capable” mainline Kernel [31].

Markets

In a July 2011 research note, VDC Research analyst Jared Weiner stated that the commercial market for Linux related software and services passed $140 million in 2010 and is expected to maintain double-digit growth through 2013. Weiner also expects Android-related services and support will be among the largest drivers of growth in the broader Linux market as interest grows among embedded developers. [32]

I'll not talk much about ”classic” Embedded Linux here, since it's already used in every imaginable kind of market segment, but I'll concentrate more on Android. That said I believe that Android is no silver bullet for every embedded problem. It's roadmap is dictated by Google which means primarily the consumer market, like mobile phones and tablets. [33] [34] This is not surprising, since Google

[3] Check the Android code below the `/external` folder and you'll see that those sources were written for `glibc`. They are still Android compatible because they refrain from using certain `glibc` features.

[4] `Bionic` supports `UNIX` and `internet sockets` and pipes. `Sockets` are typically used by the Android OS-layer (e.g. `netd`, `vold`, `init`) while components above the `system_server` typically utilize `Binder`.

[5] See `bionic/libc/docs/SYSV-IPC.TXT` in the Android sources.

[6] `Binder` is not used directly, but you must define an interface in IDL (Interface Definition Language) which is translated by `aidl()` into Java interface definitions.

[7] It's not easy to output to a graphical window with Android from C/C++ code.

can make more money in mobile than it can on the desktop with the opportunities in the mobile search market [35] and 40% of U.S. mobile users own smartphones of which 40% are running Android. [36] Besides Android there is still a huge market for custom Embedded Linux and real-time solutions.

Technically speaking (maybe with some real-time extensions in hard- and/or software and some safety/security certifications) it would make sense to use Android for more than just mobile phones if you need a sexy multi-touch user interface and an easy to use Java API. But this would mean that with every new Android release you would need to ”tailor” again and again (the price for not being mainline) and this would certainly lead to market fragmentation which does not seem to be in the interest of Google. There is already fragmentation because of security concerns with ”Google Market” so that telecom providers start rolling out their own ”safe” markets which just confuses application developers.

Despite this fact some companies use Android outside the ”classic” mobile market:

- Mercedes AMG models to get Android infotainment with apps [37]
- Renault unveils Android-based infotainment with apps: R-Link [38]
- Boeing chooses Android for 787 Dreamliner's entertainment system [39]
- Using Android in Industrial Automation [40]

Conclusion

What the future will bring is unknown, but the future for both (Embedded) Linux and Android looks bright. Despite the arguments about Android patches going in and out of the Linux kernel Linus said:”there's still a lot of merger to be done. . . . but that eventually Android and Linux would come back to a common kernel, but it will probably not be for four to five years.” [41] Now that with the Android Mainlining Project we have a capable crew of people this might happen even earlier.

You can see Android more like a platform to develop applications than an operating system and like this it might be an interesting choice for projects which require touch and a nifty GUI. Beware though that it might be close to impossible for you to influence Google's roadmap if you are not one of the high priests in the ”cathedral'. [18] It's comparable to non-mainline Linux development: If you want to go through the trouble to keep up with new releases you are on your own and need to redo your patches over and over again. We can not force Google to use a mainline Linux kernel, but making it possile to run the Android user-space over a mainline Linux kernel would certainly be an interesting option for an *Android-compatible* device.

The Android kernel patches have been the subject of extensive discussions and have, thus far, not been accepted into the mainline kernel. At the 2011 Kernel Summit, though, it was decided that this code should probably go in. Longstanding tradition says that code which is actively shipped, supported, and used should eventually be merged, even if it has problems. Nothing is certain until it happens, but we will probably see some movement in this area, perhaps for the 3.3 release. [42]

References

[1] *"2net"* http://2net.co.uk/

[2] *"Nexus Computing"* http://nexus-computing.ch

[3] *"What is Linux"*
http://www.linuxfoundation.org/what-is-linux

[4] *"Distrowatch.com"*
http://distrowatch.com/

[5] *"What is Android?"*
http://developer.android.com/guide/basics/what-is-android.html

[6] *"Android Developers"*
http://developer.android.com/index.html

[7] *"Android SDK"*
http://developer.android.com/sdk/index.html

[8] *"Celebrate 20 years of Linux"*
http://content.linuxfoundation.org/20th/

[9] *"Memorable Linux Milestones"*
http://www.dipity.com/RobertKarlBerger/Memorable-Linux-Milestones/

[10] *"Happy 20th Birthday, Linux: 10 Cool Devices That Embrace You"*
http://mashable.com/2011/09/17/happy-20th-birthday-linux/

[11] *"The Android Story"*
http://www.cubelabs.com/the-android-story.php

[12] *"The history of Android"*
http://www.dipity.com/RobertKarlBerger/The-history-of-Android/

[13] *"Wikipedia - Android (operating system)"*
http://en.wikipedia.org/wiki/Android_%28operating_system%29

[14] *"250 Million Android Devices Activated, 11 Billion Apps Downloaded"*
http://goo.gl/c0Z1q

[15] *"Android 4.0 Hacked Onto The Kindle Fire"*
http://techcrunch.com/2011/12/26/video-android-4-0-hacked-onto-the-kindle-fire/

[16] *"Linux vs. Android kernels"*
http://www.dipity.com/robertkarlberger1/Linux-vs-Android-kernels/

[17] *"List of Ubuntu releases"*
http://en.wikipedia.org/wiki/List_of_Ubuntu_releases

[18] *"The Cathedral and the Bazaar"*
http://catb.org/~esr/writings/homesteading/

[19] *"GNU General Public License, version 2"*
http://www.gnu.org/licenses/gpl-2.0.html

[20] *"Open Source Licenses"*
http://www.opensource.org/licenses/index.html

[21] *"Android Open Source Project - Licenses"*
http://source.android.com/source/licenses.html

[22] *"Google Holds Honeycomb Tight"*
http://www.businessweek.com/technology/content/mar2011/tc20110324_269784.htm

[23] *"Google Refuses to Release Android Honeycomb Source For Now"*
http://www.dailytech.com/Google+Refuses+to+Release+Android+Honeycomb+Source+For+Now/article21222.htm

[24] *"Open Handset Alliance"*
http://www.openhandsetalliance.com/index.html

[25] *"Is Android Open?"*
http://www.wired.com/epicenter/2010/10/is-android-open/

[26] *"Android Chief: Were Still Open, Dammit"*
http://www.wired.com/gadgetlab/2011/04/android-google-andy-rubin-open/

[27] *"Amazons Kindle Tablet Is Very Real. Ive Seen It, Played With It."*
http://techcrunch.com/2011/09/02/amazon-kindle-tablet/

[28] *"Tagged #Facebook phone"*
http://androidandme.com/tag/facebook-phone/

[29] *"Top 5 predictions for 2012"*
http://www.technewsworld.com/story/73955.html

[30] *"Filesystem Hierarchy Standard"*
http://refspecs.linuxfoundation.org/fhs.shtml

[31] *"Android Mainlining Project"*
http://elinux.org/Android_Mainlining_Project

[32] *"Engineers Guide to Embedded Linux and Android"*
http://eproductalert.com/digitaledition/embedded_linux/2012/Engineers%20Guide%20to%20Embedded%20Linux%20and%20Android.pdf

[33] *"Android Roadmap"*
https://sites.google.com/a/android.com/opensource/roadmap

[34] *"Android@Home"*
http://www.androidathome.com/

[35] *Gomez-Barroso, Jose Luis; Feijoo, Claudio; Compano, Ramon; , "Opportunities in the Mobile Search Market," Computer , vol.44, no.11, pp.83-85, Nov. 2011 doi: 10.1109/MC.2011.347*
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6072571&isnumber=6072553

[36] *"40 Percent of U.S. Mobile Users Own Smartphones; 40 Percent are Android"*
http://goo.gl/gN90I

[37] *"Mercedes AMG models to get Android infotainment with apps"*
http://goo.gl/UeuWZ

[38] *"Renault unveils Android-based infotainment with apps: R-Link"*
http://goo.gl/m6XMa

[39] *"Boeing chooses Android for 787 Dreamliner's entertainment system"*
http://goo.gl/fYXpq

[40] *"Using Android in Industrial Automation"*
http://android.serverbox.ch/

[41] *"Linus Torvalds on Android, the Linux fork"*
http://www.zdnet.com/blog/open-source/linus-torvalds-on-android-the-linux-fork/9426?tag=search-results-rivers;item5

[42] *"Linux Weather Forecast"*
http://www.linuxfoundation.org/news-media/lwf

Author

Robert Berger consults and trains people all over the globe on a mission to help them create better embedded software. His specialties are trainings and consulting in the broad field of Embedded Software from small Real-time Systems to multi-core Embedded Linux.