Embedded in 2010: An End to the Entropy?

Matt Asay
COO, Canonical
...and where “smart” meets bankruptcy
Remember these?

Board Support Package (BSP)

The BSP contains routines for initializing and controlling:

- Interface with boot and shutdown software
- Establish virtual address map for onboard I/O
- Interface with the interrupt controller
- Provide default handlers for error-signaling interrupt
- Interface with the PCI controller
- Interface with the system time (tick timer)
The Past

- ARMv1 – ARMv3
  - Obsolete: Not powerful enough to run Linux.

- ARMv4
  - Some very limited Linux BSP's and some level of support in Debian without X. Heavily used in the embedded space with RTOS's

- ARMv5-ARMv6
  - Powerful enough for phones and some smart phones. Limited Linux, and other RTOS's used.
As we grew up...
Today

ARMv7 + Intel Atom + ...

- Powerful enough for smart phones and general-purpose computing, desktop, note/netbooks, and low-power servers

- Need a well-supported general-purpose operating system to support use cases
In summary: BSPs are just the beginning now

Basic SoC enablement is no longer good enough

- Low-power chips required bare metal to get enough power to run a few apps
- Increasingly powerful chips mean OEMs/ODMs want general purpose OSes

The race is on to create and fill app stores, not merely a PIM suite

- From a handful to dozens to hundreds to thousands to... (?) of apps
- But is there something bigger than an app store?
Yes. It's called “The Internet”
Don't be confused by device proliferation
Why so much device divergence?

- Cheaper to build
  - Lower cost of success... and failure

- Huge pay-out for getting it right
  - iPhone, RAZR, Blackberry Pearl, etc.
This money leads to...

Lots of exploration...

- “Embedded” looks more and more like a general-purpose device OS tied to an even more general-purpose cloud OS

Complete guesswork...

- 50%+ of under-15s will use touchscreen PCs by 2015 (Gartner)

And breathless exaggeration
Cloud + Device = Gold Mine

Single-purpose gadgets ->
Single-purpose apps w/ general-purpose OS ->
General-purpose cloud running Linux
OS increasingly common between different form factors
Linux increasingly the common platform
Long ago Linux won the embedded market.
Mobile market next?

![Mobile OS 2009 market share Gartner](chart)

- Symbian: 47%
- Research In Motion: 20%
- iPhone OS: 14%
- Microsoft Windows Mobile: 9%
- Linux: 5%
- Android: 4%
- WebOS: 1%
- Other OSs: 0%
Why Linux?

• Device+Services platforms is expensive. Linux can help
  – Lowers the cost of building/maintaining platforms
  – Built-in development community
  – Enables pooled effort to innovate the platform
    • Embrace and extend existing platforms
• New center of real mobile innovation, from RT to memory management to graphics performance
• Allows solution providers to focus on added value (UI, services integration, etc.)
That's how Linux **should** be...but not how it is
Linux can be the answer

- App developers want to build and sell new apps; OEMs want to create and sell new SKUs
  - Both can be done with Linux. But we must not go down closed code lines
- Android, e.g., makes life easier for app developers but harder for OEMs (i.e., can't differentiate) – same old Windows problem

Google Draws Criticism for Android Changes

By Carmen Nobel  03/18/10 - 11:02 AM EDT  2 Comments | Add Cor
...but not this way

- Android is a closed, quasi-Linux community
  - Google code line ~8 months ahead of public SDK, with certification/code review by Google
  - Hurts device ODMs/OEMs who must invest in Android, not Linux, skills + QA
  - Speed at expense of the upstream community

"...due to time and resource constraints on [Google's] side, and an inability on our side to make changes that would require core Android system changes, nothing ever got accomplished."
  Greg Kroah-Hartman

"You can't ship a nice phone without customizing the kernel to the hardware, and these changes are what makes the code for a platform divert from the mainline tree."
  Chris DiBona

Can we converge on an open-source, open-process Linux?
Linux is Linux is Linux...right?

So why aren't we better coordinating upstream efforts with enablement?
A BSP for every user environment is crazy talk

Hurtful to community (upstream)

Hurtful to vendors (reinventing + maintaining the wheel)
“They make a desert and call it peace.”

- Tacitus
Wouldn't Linux mobile look better like this?

User Environments

Android
Chrome OS
palm webOS™
MeeGo™
LiMo
D-Link Boxee – An example of the possibilities

- D-Link Boxee Media Player, *Won multiple 'Best of CES 2010' awards*
  - Nvidia Tegra 2 (ARM Cortex A9)
  - Built on **Ubuntu Core**
- Great example of **accelerated product development** because both SiP and ISV were already using Ubuntu Core
- Don't get me wrong: Canonical has work to do (UDS focus on power, optimization, device tree, etc.)
  - But the idea is to build on a common Linux and focus customization “up the stack”
“Everything that rises must converge.”
Flannery O'Connor (via Pierre Teilhard De Chardin)