Can Linux Cope with Diversity?

David A Rusling

CTO, Linaro

Linux Plumbers, September 2011

WARNINGS

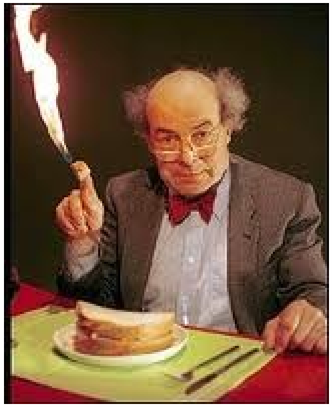
- I have a very **mobile** view of the world
  - Aside – embedded is an out of date concept (everything is connected)
- I helped create **Linaro** , so I'm a bit biased
- Architectures == {Alpha, ARM}

History

- ARM born in 1990 – a joint venture between Acorn, VLSI and Apple
- Innovative step was its licensing model
- ARM grew rapidly, driven by the mobile and embedded industries
- Linaro was formed to allow the ARM partnership to better interact with open source

Throwing Hardware at the Wall

- ARM has (traditionally) not constrained the system architecture
- Each ARM licensee has their own platform
- Result is hardware **diversity**



The View from the Silos

- ARM Linux engineers have (traditionally) worked on their own SoC
- Deadlines in the embedded world are very short (~4 months not uncommon)
- No excuses, but a lot of naivety about open source software, its processes and especially around copyright and patents
- **Fragmentation** was inevitable
- This gives Linaro it's prime mission - **consolidation**

Collaboration

- Linaro's *other* mission, without which we can achieve nothing
- ARM is a collective of many, many companies
- They have recognized the need to **collaborate** in Open Source
- OK, so what **problems** should we be solving?

Linux, Microsoft and the Desktop

A long time ago, in a galaxy far, far away...

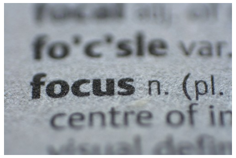
- Microsoft was the empire
- Linux – rebel child of the internet age and GPL
- Very successful but settling into a comfortable middle age
- Laptops ship because Windows works, not because Linux does (chasing the hardware taillights)
- 'Classic' Linux is not very diverse as it relies on the PC architecture
- The desktop is not very interesting / relevant
  - Server and mobile are...

Server

- ARM server will be different
- Client versus server load balancing versus communication bandwidth
- Long term kernel impacts (scheduling, power management, load balancing, huge memory?...)
- Mobile 'thinking' will apply to Server and vice versa

Android

- Since 2007, has become the **unifying** response to iPhone and the **de facto mobile Linux distribution**
- Very product and post-PC world focused
- **Fork the kernel**, release, rebase and repeat
  - Work with lead partner on releases
- Highly integrated graphics and multimedia supporting novel interactions
- **Obsessed** with battery lifetime
- Remember, in order to ship these products, **Linux must work** on them...



Supporting Diversity

Focus on:

- Collaboration
- Kernel consolidation
- Boot architecture
- Memory management
- Power Management

...always with low power and a long battery life

Collaboration

- This **habit** of collaboration (in the ARM community) is growing, but we still need help and support from the wider community
- There needs to be **more ARM Kernel maintainers** and they need to collaborate more often
  - Code reviews etc
- **Linaro** can help here, we're an open organization, we can join support initiatives

Boot Architecture

- Loosely, everything that happens before the kernel
- Standards, such as UEFI
  - Influenced by the Linux community?
- Communicating system information to the kernel
  - ACPI versus device tree (versus code)

Post-Platform Era?

- Will there ever be a single ARM platform?
- Possible, but will happen over time
- Already have **many platforms**, for example TI OMAP, iMX ...
- How should this be handled in the kernel?

Kernel Consolidation

- /arm should contain architectural and SoC platform specific code, otherwise should be in common kernel places
- Look for **patterns** and make generic
- Aim for a **generic kernel** – there are many interesting problems to solve along the way
- How does this fit with other architectures?

arm-soc

- Linaro is supporting the ARM subarchitecture maintained tree (arm-soc)
- Goals:
  - Support Russell King and the ARM Linux kernel maintainers – it's all about engineering efficiency
  - Vehicle for consolidation, moving common code to common places (example is the GPIO driver code)

Power Management

- Thermal framework – being able to gather thermal information uniformly
- User space governors with SoC specific information / code versus generic mechanisms
- Good debug and instrumentation essential to tune / balance

The Post-Desktop Era

- Does an ARM based desktop system look like an x86 based desktop?
- Gnome 3 and Unity both show the right direction of mixed mobile and traditional desktop interaction
- Whatever it looks like, the future will be based on integrated graphics and multimedia acceleration
  - ... with a > 10 hour battery life

Graphics and Multimedia

- Need well integrated, efficient (processing and power) graphics and multimedia
  - That means efficient buffer management, capability handling
- Kernel mechanisms do not (currently) cope with ARM systems
  - Weakly ordered memory hierarchies
  - System MMUs
- Many solutions out there, some clashing, mostly not upstream

Memory Management

- Working on
  - CMA (Contiguous Memory Allocator),
  - dma_map_ops,
  - common GPU IOMMU
- Future of buffer allocation?
  - Buffer descriptors (file descriptors versus other)
- UCM (use case management)
  - As used in sound (alsa); makes sense here?

Back to the Question...

Assuming:

- Linux community is robust and adaptable
- That the ARM community organizes itself (reasonably) efficiently

The answer is **yes**, but...

Social Engineering

- Tension between product releases and kernel engineering lead times is critical
  - What difference can be coped with (4 months of product release time versus 1 or 2 years of kernel ~~engineering~~ arguing time)?
  - There's a **social engineering** problem to be solved
- Go forth and fork

Scaling

Scaling solutions across many diverse platforms:

- Moving from works on (each) one to works on all
- Generic kernel subsystems supporting diverse systems
- From server to mobile (and all points in between)

Crossing the Borders

- System knowledgeable user space agents using kernel subsystems
  - Graphics, power governors
- All of these are complex and difficult and need to cross the kernel / user 'border'
  - Sometimes, many times in many places
- **Where** and **how** do we have the discussion (and make decisions)

Questions?

www.linaro.org

Title

- Bullet
- Another bullet