CE Workgroup

Linaro

Kernel Mainline Status of Mobile Chipsets

September, 2015

Tim Bird

LF CE Workgroup

Agenda

Linaro

- Big-picture status
- Device mainlining project
- Activities
- Where can I learn more?
- How can I participate?
- Discussion

Agenda

Linaro

- Big-picture status
- Device mainlining project
- Activities
- Where can I learn more?
- How can I participate?
- Discussion

Big picture status

Linaro

- Most mobile devices have between 1 and 3 million lines of code out-of-tree
  - This is for shipping products (v3.4 era)
  - Mobile devices (and many embedded products) are 3 years and 20 versions behind mainline
    - Referred to as “Version Gap”
- End users and product developers can’t use mainline kernels on their hardware
  - Low interaction with mainline from device manufacturers (few updates and contributions)
  - Ghetto-ization of patches for mobile devices

Out-of-tree SoC code

Linaro

Company	SOC	Files	Insertions	Deletions
LG	Msm	5775	2.616M	40K
Motorola	Msm	4490	1.795M	40K
Samsung	Exynos	2877	1.100M	51K
Samsung	Msm	6096	3.105M	53K
Sony	Msm	4625	1.784M	41K
Sony	Mediatek	3689	1.935M	7K
Acer	Mediatek	3122	1.411M	6K
Asus	Atom	7351	2.163M	22K
Huawei	Hisilicon	5082	2.659M	43K

What's the *bigger* problem?

Linaro

- For manufacturers
  - Working with Linux is hard!
    - Sony Mobile has 1100 developers who made a patch to the kernel in the last 3 years
  - Device manufacturers don't participate in open source
    - Institutional barriers
      - Don't recognize benefits
      - Don't know how
    - Version gap
- For users
  - Devices are abandoned
  - No long-term support path for their hardware

What's the *bigger* problem?

Linaro

- For manufacturers
  - Working with Linux is hard!
    - Sony Mobile has 1100 developers who made a patch to the kernel in the last 3 years
  - Device manufacturers don't participate in open source
    - Institutional barriers
      - Don't recognize benefits
      - Don't know how
    - Version gap → Is both a cause and an effect of device mainlining problems
- For users
  - Devices are abandoned
  - No long-term support path for their hardware

Path of broken dreams

Linaro

Mainline

Google

SoC Vendor

Device Manufacturer

End User

Version Gap

Cycle of frustration

Linaro

Mainline

Google

SoC Vendor

Device Manufacturer

End User

Version Gap

Google

SoC Vendor

...

Agenda

Linaro

- Big-picture status
- Device mainlining project
- Activities
- Where can I learn more?
- How can I participate?
- Discussion

Device mainlining project

Linaro

- Joint project between Linaro and Linux Foundation
- Make it easier for developers to upstream currently out-of-tree code
  - Determine obstacles to mainlining
  - Reduce or eliminate those obstacles

Device mainlining project

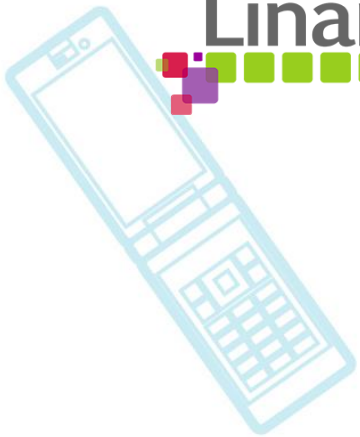
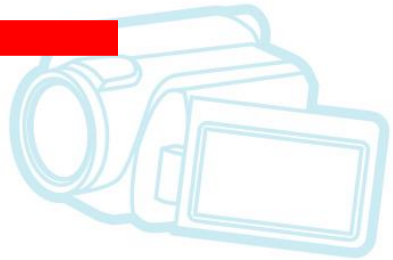
Linaro

- Identification and education:
  - Obstacles Survey
  - Presentations and white paper
  - Training
- Technical
  - Out-of-tree code analysis
  - Projects to address specific technical issues
  - Tools for new contributors
- SIGs/BOFs to discuss issues

Agenda

Linaro

- Big-picture status
- Device mainlining project
- Activities
- Where can I learn more?
- How can I participate?
- Discussion

Activities

Linaro

- Recent past:
  - Survey of corporate developers who don't contribute
  - "Obstacles" Talk and White Paper
    - White paper at: http://elinux.org/images/e/ed/Overcoming-Obstacles-to-Mainlining-White-Paper-version-0.9.pdf
    - LWN.net article at: https://lwn.net/Articles/647524/

Technical Analysis

Linaro

- SoC out-of-tree code analysis
  - Upstream-analysis-tools
    - Set of tools to categorize diffs between production source trees and mainline
    - http://elinux.org/Phones_Processors_and_Download_Sites
    - https://github.com/tbird20d/upstream-analysis-tools
  - Purpose is to find areas to work on
- Mainline technical areas of focus
  - http://elinux.org/Kernel_areas_of_focus_for_mainlining
    - Has notes for major areas of out-of-tree code, and ideas for projects to work on

Big problem areas

Linaro

Area	Insertions range
Mach-xxx	347K – 417K
Media	120K – 360K
Video	37K – 346K
Wireless	80K – 250K
Sound	74K – 240K
Input	51K – 238K
Camera	50K – 210K
GPU	36K – 172K
Power	44K – 94K

Qcom overview

Linaro

Area/directory	# of lines insertions
Mach-msm	375K
Video	200K
Sound	200K
Media	120K
USB	93K
DTS	55K
Gpu	52K
Input	50K
Camera	50K
Total	1600K

Specific projects by Linaro, CE Workgroup

Linaro

- Wireless drivers
  - Help mature the mainline broadcom wireless driver
    - CEWG project to backport brcm80211 to 3.14
    - See http://elinux.org/Support_mainline_Broadcom_wireless_driver_on_an_Android_platform
- USB
  - Integration with charger
  - Extcon for USB pins not connected to controller hardware

Other technical areas

Linaro

- http://elinux.org/Kernel_areas_of_focus_for_mainlining
- Sensors – promote the use of IIO
- Charging – need kernel framework for this
  - Lots of vendor charging code is in userspace now
- NFC/GPS/Bluetooth (and other things with weird UART-based drivers)
  - UART slave (can someone explain this to me?)

Other areas

Linaro

- Some institutional barriers, as well as process issues
  - http://elinux.org/Mainlining_improvement_ideas
- Use this page to help describe:
  - Obstacles to overcome
  - New ideas for tools
  - New ideas for management education
- Training resources:
  - http://elinux.org/Kernel_Mainlining

Agenda

Linaro

- Big-picture status
- Device mainlining project
- Activities
- Where can I learn more?
- How can I participate?
- Discussion

Where can I learn more?

Linaro

- Main project web site:
  - http://elinux.org/CE_Workgroup_Device_Mainlining_Project
- Mailing list:
  - http://lists.linuxfoundation.org/mailman/listinfo/device-mainlining

Agenda

Linaro

- Big-picture status
- Device mainlining project
- Activities
- Where can I learn more?
- How can I participate?
- Discussion

How can I participate?

Linaro

- Identify deficiencies
  - Run upstream-analysis-tools yourself
- Put out-of-tree code into mainline
- Fix upstream code so it is product-grade
- Document benefits of mainlining
- Write and enhance tools to make mainlining easier
- Add documentation for newcomers
- Suggest more ideas

Agenda

Linaro

- Big picture status
- Device mainlining project
- Activities
- Where can I learn more?
- How can I participate?
- Discussion

Linaro

DISCUSSION

Possible discussion points

Linaro

- Technical issues:
  - What other areas are deficient?
  - Are there solutions being worked on?
- Non-technical issues:
  - Convincing management to contribute
  - Big problem seems to be multi-OS code
  - Would like examples of code reduction from conversion to mainline drivers (anyone??)
  - Would like examples of maintenance reduction from conversion to mainline drivers (next project)

Linaro

- Past here are slides for possible discussion points
  - Survey, samsung/qualcomm comparison, kernel contribution stats
  - where companies get stuck, DT review

Survey

Linaro

- Conducted survey in September 2014
- To determine perceived obstacles
- Top obstacles (from survey):

Obstacle	How many agreed?
Older kernel version	54%
Depends on other code not upstream	50%
It’s too hard	45%
Could not test	41%
Patch not good enough	35%
Employer does not provide time	40%
Afraid of rejection	33%

Samsung Highlights

Linaro

Area/directory	# of lines insertions (msm)	# of lines insertions (exynos)
Mach-xxxx	347K	89K
Media	364K	163K
Video	346K	176K
Sound	239K	86K
Wireless	251K	80K
Firmware	242K	101K
Input	238K	51K
Camera	121K	1K
USB	117K	35K
DTS	99K	0K
Gpu	53K	172K
Total	3105K	1100K

Kernel contribution notes

Linaro

- Contributions by different companies

Author email domain	commits	Commiters (since 3.4)
Sony[me] *(sonymobile)*	53	14
Lge.com	565	11
Huawei	1220	71
Qualcomm|Codeaurora	1349	46
Moto	1035	15
Free-electrons	2333	9
Samsung	7031	160
Intel	17374	469

Results from: git log v3.4.. –author=<expr> --format=%ae | sort | uniq | wc -l

Where companies get stuck

Linaro

- Discussion from SIG meeting in March
  - Incentives
  - Product treadmill mismatch with mainlining
    - Product teams are too busy to learn OSS methods and contribute
  - Technical issues

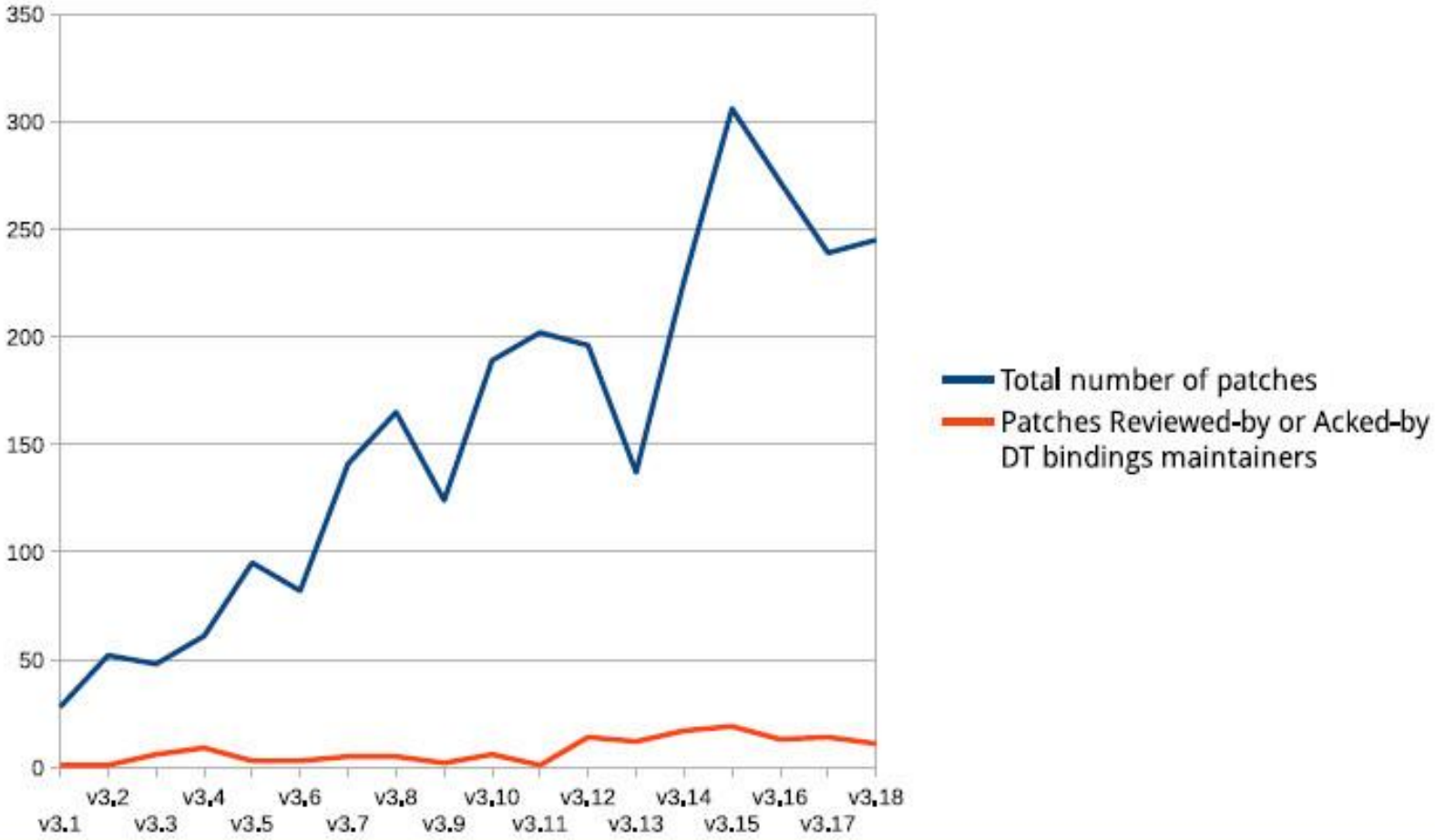
Technical/Community issues

Linaro

- Devicetree binding approval bottleneck
  - Thomas Petazzoni's slides (next page)
    - From "Device Tree Stable ABI – a Fairy Tale", presented at ELC
- Slow (non-responsive) maintainers
  - Example: hwspinlock, rpmsg, rtc
- Framework issues
  - Example: upstream USB - state machine doesn't know about charging. Obviously can't be used for real products.

Enough review?

- Stability of the system call ABI is achieved by careful review of the proposed changes.
- What amount of review do we have for DT bindings?



Ideas

Linaro

- DT staging / SoC support in staging?
- Maintainer assistance
  - Help overloaded or slow maintainers
- Specific frameworks or sub-systems
  - Wireless, USB gadget (already discussed)
  - Media, video, sound, input
  - NFC, bluetooth
  - Low-level SoC support (mach-xxxx)
    - Regulators, clocks, resets, gpio, pinctrl, inter-processor communication, power management