Device Mainlining Project

August, 2015

Tim Bird
LF CE Workgroup

Mark Brown
Linaro Kernel Technical Lead

Kate Stewart
Linux Foundation
Device Mainlining

- What’s the problem?
- What is this project?
- What’s going on?
- Where can I learn more?
- How can I participate?
- Discussion
Device Mainlining

- What’s the problem?
- What is this project?
- What’s going on?
- Where can I learn more?
- How can I participate?
- Discussion
What’s the problem?

- 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

<table>
<thead>
<tr>
<th>Company</th>
<th>SOC</th>
<th>Files</th>
<th>Insertions</th>
<th>Deletions</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG</td>
<td>Msm</td>
<td>5775</td>
<td>2.616M</td>
<td>40K</td>
</tr>
<tr>
<td>Motorola</td>
<td>Msm</td>
<td>4490</td>
<td>1.795M</td>
<td>40K</td>
</tr>
<tr>
<td>Samsung</td>
<td>Exynos</td>
<td>2877</td>
<td>1.100M</td>
<td>51K</td>
</tr>
<tr>
<td>Samsung</td>
<td>Msm</td>
<td>6096</td>
<td>3.105M</td>
<td>53K</td>
</tr>
<tr>
<td>Sony</td>
<td>Msm</td>
<td>4625</td>
<td>1.784M</td>
<td>41K</td>
</tr>
<tr>
<td>Sony</td>
<td>Mediatek</td>
<td>3689</td>
<td>1.935M</td>
<td>7K</td>
</tr>
<tr>
<td>Acer</td>
<td>Mediatek</td>
<td>3122</td>
<td>1.411M</td>
<td>6K</td>
</tr>
<tr>
<td>Asus</td>
<td>Atom</td>
<td>7351</td>
<td>2.163M</td>
<td>22K</td>
</tr>
<tr>
<td>Huawei</td>
<td>Hisilicon</td>
<td>5082</td>
<td>2.659M</td>
<td>43K</td>
</tr>
</tbody>
</table>
What’s the *bigger* problem?

- **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?

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
Path of broken dreams

Mainline

Google

SoC Vendor

Device Manufacturer

Version Gap

End User
Cycle of frustration

Mainline

Google

SoC Vendor

Device Manufacturer

Version Gap

Google

SoC Vendor

End User

...
Device Mainlining

- What’s the problem?
- What is this project?
- What’s going on?
- Where can I learn more?
- How can I participate?
- Discussion
What is this project?

- “Device Mainlining” project
- 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

- 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
Device Mainlining

- What's the problem?
- What is this project?
- What's going on?
- Where can I learn more?
- How can I participate?
- Discussion
What’s going on?

Recent past:
- Survey of corporate developers who don’t contribute
- “Obstacles” Talk and White Paper
  - LWN.net article at: https://lwn.net/Articles/647524/
Technical Analysis

- 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

<table>
<thead>
<tr>
<th>Area</th>
<th>Insertions range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mach-msm</td>
<td>347K – 417K</td>
</tr>
<tr>
<td>Media</td>
<td>120K – 360K</td>
</tr>
<tr>
<td>Video</td>
<td>37K – 346K</td>
</tr>
<tr>
<td>Wireless</td>
<td>80K – 250K</td>
</tr>
<tr>
<td>Sound</td>
<td>74K – 240K</td>
</tr>
<tr>
<td>Input</td>
<td>51K – 238K</td>
</tr>
<tr>
<td>Camera</td>
<td>50K – 210K</td>
</tr>
<tr>
<td>GPU</td>
<td>36K – 172K</td>
</tr>
<tr>
<td>Power</td>
<td>44K – 94K</td>
</tr>
</tbody>
</table>
# Qcom overview

<table>
<thead>
<tr>
<th>Area/directory</th>
<th># of lines insertions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mach- msm</td>
<td>375K</td>
</tr>
<tr>
<td>Video</td>
<td>200K</td>
</tr>
<tr>
<td>Sound</td>
<td>200K</td>
</tr>
<tr>
<td>Media</td>
<td>120K</td>
</tr>
<tr>
<td>USB</td>
<td>93K</td>
</tr>
<tr>
<td>Dts</td>
<td>55K</td>
</tr>
<tr>
<td>Gpu</td>
<td>52K</td>
</tr>
<tr>
<td>Input</td>
<td>50K</td>
</tr>
<tr>
<td>Camera</td>
<td>50K</td>
</tr>
<tr>
<td>Total</td>
<td>1600K</td>
</tr>
</tbody>
</table>
Specific projects by Linaro, CE Workgroup

- **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 areas

- 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
Device Mainlining

- What’s the problem?
- What is this project?
- What’s going on?
- Where can I learn more?
- How can I participate?
- Discussion
Where can I learn more?

- **Main project web site:**
  - http://elinux.org/CE_Workgroup_Device_Mainlining_Project

- **Mailing list:**
  - http://lists.linuxfoundation.org/mailman/listinfo/device-mainlining
Device Mainlining

- What’s the problem?
- What is this project?
- What’s going on?
- Where can I learn more?
- How can I participate?
- Discussion
How can I participate?

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

- What’s the problem?
- What is this project?
- What’s going on?
- Where can I learn more?
- How can I participate?
- Discussion
DISCUSSION
Survey

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

<table>
<thead>
<tr>
<th>Obstacle</th>
<th>How many agreed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Older kernel version</td>
<td>54%</td>
</tr>
<tr>
<td>Depends on other code not upstream</td>
<td>50%</td>
</tr>
<tr>
<td>It’s too hard</td>
<td>45%</td>
</tr>
<tr>
<td>Could not test</td>
<td>41%</td>
</tr>
<tr>
<td>Patch not good enough</td>
<td>35%</td>
</tr>
<tr>
<td>Employer does not provide time</td>
<td>40%</td>
</tr>
<tr>
<td>Afraid of rejection</td>
<td>33%</td>
</tr>
</tbody>
</table>
## Samsung Highlights

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

### Contributions by different companies

<table>
<thead>
<tr>
<th>Author email domain</th>
<th>commits</th>
<th>Committers (since 3.4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sony[me] (sonymobile)</td>
<td>53</td>
<td>14</td>
</tr>
<tr>
<td>Lge.com</td>
<td>565</td>
<td>11</td>
</tr>
<tr>
<td>Huawei</td>
<td>1220</td>
<td>71</td>
</tr>
<tr>
<td>Qualcomm</td>
<td>Codeaurora</td>
<td>1349</td>
</tr>
<tr>
<td>Moto</td>
<td>1035</td>
<td>15</td>
</tr>
<tr>
<td>Free-electrons</td>
<td>2333</td>
<td>9</td>
</tr>
<tr>
<td>Samsung</td>
<td>7031</td>
<td>160</td>
</tr>
<tr>
<td>Intel</td>
<td>17374</td>
<td>469</td>
</tr>
</tbody>
</table>

*Results from: git log v3.4.. --author=<expr> --format=%ae | sort | uniq | wc -l*
Where companies get stuck

- 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

• 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

- 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-xxxxx)
    - Regulators, clocks, resets, gpio, pinctrl, interprocessor communication, power management