LTSI Project update

Long Term Support Initiative

Tsugikazu SHIBATA, NEC
23, Oct. 2017
Embedded Linux Conference Europe
Hilton Prague
agenda

• Kernel statistics and process
• History of LTSI and learned in 6 years
• Further steps for maintaining kernel long term
Who am I

• Tsugikazu SHIBATA, NEC
• Founder and project lead of LTSI
• LTS/LTSI Advocate

• Board member of Linux Foundation
• Involved with Linux community since 2.4
Linux = Open Source project

• Linux is one of the most successful Open Source project

• Continue growing in 26 years; expanding adoption for new area;
  – IT enterprise, Cloud, Network, Smart Phone, Robotics, Embedded, IoT and many others

• Developing and delivering under GPLv2
Developed by the community

• Participating ~1700 developer, ~230 companies every releases
• Growing yearly 1.5M lines of code, 4000 files increased
• 26 Years of history
• Maintainers have great skill to manage the subsystem and professional knowledge of its area of technologies
Status of Latest Linux Kernel

- Latest released Kernel: 4.13
  - Released: Sep 3rd, 2017
  - Lines of code: 24,767,008 (+596,148)
  - Files: 60,543 (+737)
  - Developed period: 63 days from 4.12
- Current Stable Kernel: 4.13.9
- Current development kernel: 4.14-rc5
Kernel release cycle

- Release cycle: 65 ~ 70 days, 5~6 releases/year

<table>
<thead>
<tr>
<th>Version</th>
<th>Release</th>
<th>Rel. span</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.19</td>
<td>2015-2-9</td>
<td>64</td>
</tr>
<tr>
<td>4.0</td>
<td>2015-4-12</td>
<td>62</td>
</tr>
<tr>
<td>4.1</td>
<td>2015-6-22</td>
<td>71</td>
</tr>
<tr>
<td>4.2</td>
<td>2015-8-30</td>
<td>69</td>
</tr>
<tr>
<td>4.3</td>
<td>2015-11-2</td>
<td>64</td>
</tr>
<tr>
<td>4.4</td>
<td>2016-1-10</td>
<td>68</td>
</tr>
<tr>
<td>4.5</td>
<td>2016-3-14</td>
<td>64</td>
</tr>
<tr>
<td>4.6</td>
<td>2016-5-15</td>
<td>63</td>
</tr>
<tr>
<td>4.7</td>
<td>2016-7-24</td>
<td>70</td>
</tr>
<tr>
<td>4.8</td>
<td>2016-10-2</td>
<td>70</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version</th>
<th>Release</th>
<th>Rel. span</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.9</td>
<td>2016-12-11</td>
<td>70</td>
</tr>
<tr>
<td>4.10</td>
<td>2017-02-09</td>
<td>60</td>
</tr>
<tr>
<td>4.11</td>
<td>2017-04-30</td>
<td>80</td>
</tr>
<tr>
<td>4.12</td>
<td>2017-07-02</td>
<td>63</td>
</tr>
<tr>
<td>4.13</td>
<td>2017-09-03</td>
<td>63</td>
</tr>
<tr>
<td>4.14</td>
<td>2017-??</td>
<td></td>
</tr>
</tbody>
</table>
Linux development policy

- Upstream is only the place to accept the patches
  - Reviewed by skilled maintainers
  - Tested with other proposals to confirm no conflicts
  - Well coordinated development process for over thousands of developers
Linux Development process

- Just after the release of 4.n, two weeks of merge window will be opened for proposal of new features.
- After 2 weeks of merge window, -rc1 will be released and the stabilization will be started.
- 4.n+1 will be released when it becomes reasonably stable by some of -rcX released.
Linux Source Code Growth

- Increasing 0.3ML/Version, 1.5ML/year
Rapid Release cycle of Linux

• Yearly more than 5 times of chance to merge the code into upstream.
  – Other project maybe 6 month release cycle that is 2 times/year
• Lot’s of chance to merge new code into upstream
• So many choice to use for newer products and need deeper knowledge to pick right version
• Recommended branch for users who want the most recent stable kernel
• 3 part version like 4.n.m
• Contain small and critical fixes for security problems or significant regressions discovered in a latest development version
• Becomes “End Of Life” when next stable kernel were released
Status of Latest Linux Kernel Again

- Latest released Kernel : 4.13
- Current Stable Kernel : 4.13.7
- Current development kernel : 4.14-rc5
LTS: Long Term Stable Kernel

• Extended maintenance period for stable kernel
• Kernel tree continue to back port bug and Security fixes for more long term
• Pick one version per year and maintain 2 years
Why LTS?

• Only the tree get fixes from the community
• In the real use case, tested/confirmed kernel is important, less important for new features
• Fixes will be released # of times and should be applied frequently, Security/Bug fixes are being more important
• Bugs found in LTS should be reported and fixed in upstream
## Current LTS versions

<table>
<thead>
<tr>
<th>Version</th>
<th>Maintainer</th>
<th>Released</th>
<th>Projected EOL</th>
<th>Years</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.9</td>
<td>Greg Kroah-Hartman</td>
<td>2016-12-11</td>
<td>Jan, 2019</td>
<td>2</td>
</tr>
<tr>
<td>4.4</td>
<td>Greg Kroah-Hartman</td>
<td>2016-01-10</td>
<td>Feb, 2022</td>
<td>6</td>
</tr>
<tr>
<td>4.1</td>
<td>Sasha Levin</td>
<td>2015-06-21</td>
<td>May, 2018</td>
<td>3</td>
</tr>
<tr>
<td>3.16</td>
<td>Ben Hutchings</td>
<td>2014-08-03</td>
<td>Apr, 2020</td>
<td>6</td>
</tr>
<tr>
<td>3.10</td>
<td>Willy Tarreau</td>
<td>2013-06-30</td>
<td>Oct, 2017</td>
<td>4</td>
</tr>
<tr>
<td>3.2</td>
<td>Ben Hutchings</td>
<td>2012-01-04</td>
<td>May, 2018</td>
<td>6</td>
</tr>
</tbody>
</table>

[https://www.kernel.org/category/releases.html](https://www.kernel.org/category/releases.html)
LTS includes large number of fixes

- 600 – 700 fixes included in a Stable release
- LTS include several thousands of fixes

As of 2017/10/15

<table>
<thead>
<tr>
<th>Version</th>
<th>FROM-TO</th>
<th>#Commits</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>3.2.94</td>
<td>8105</td>
</tr>
<tr>
<td>3.3</td>
<td>3.3.8</td>
<td>698</td>
</tr>
<tr>
<td>3.4</td>
<td>3.4.113</td>
<td>5929</td>
</tr>
<tr>
<td>3.5</td>
<td>3.5.7</td>
<td>816</td>
</tr>
<tr>
<td>3.6</td>
<td>3.6.11</td>
<td>757</td>
</tr>
<tr>
<td>3.7</td>
<td>3.7.10</td>
<td>718</td>
</tr>
<tr>
<td>3.8</td>
<td>3.8.13</td>
<td>996</td>
</tr>
<tr>
<td>3.9</td>
<td>3.9.11</td>
<td>746</td>
</tr>
<tr>
<td>3.10</td>
<td>3.10.107</td>
<td>6564</td>
</tr>
<tr>
<td>3.11</td>
<td>3.11.10</td>
<td>677</td>
</tr>
<tr>
<td>3.12</td>
<td>3.12.70</td>
<td>7342</td>
</tr>
<tr>
<td>3.13</td>
<td>3.13.11</td>
<td>903</td>
</tr>
<tr>
<td>3.14</td>
<td>3.14.79</td>
<td>4977</td>
</tr>
<tr>
<td>3.15</td>
<td>3.15.10</td>
<td>703</td>
</tr>
<tr>
<td>3.16</td>
<td>3.16.49</td>
<td>7278</td>
</tr>
<tr>
<td>3.17</td>
<td>3.17.8</td>
<td>884</td>
</tr>
<tr>
<td>3.18</td>
<td>3.18.75</td>
<td>5281</td>
</tr>
<tr>
<td>3.19</td>
<td>3.19.8</td>
<td>873</td>
</tr>
<tr>
<td>4.0</td>
<td>4.0.9</td>
<td>757</td>
</tr>
<tr>
<td>4.1</td>
<td>4.1.44</td>
<td>4629</td>
</tr>
<tr>
<td>4.2</td>
<td>4.2.8</td>
<td>903</td>
</tr>
<tr>
<td>4.3</td>
<td>4.3.6</td>
<td>618</td>
</tr>
<tr>
<td>4.4</td>
<td>4.4.92</td>
<td>5619</td>
</tr>
<tr>
<td>4.5</td>
<td>4.5.7</td>
<td>973</td>
</tr>
<tr>
<td>4.6</td>
<td>4.6.7</td>
<td>705</td>
</tr>
<tr>
<td>4.7</td>
<td>4.7.10</td>
<td>912</td>
</tr>
<tr>
<td>4.8</td>
<td>4.8.17</td>
<td>1102</td>
</tr>
<tr>
<td>4.9</td>
<td>4.9.56</td>
<td>4838</td>
</tr>
<tr>
<td>4.10</td>
<td>4.10.17</td>
<td>1136</td>
</tr>
<tr>
<td>4.11</td>
<td>4.11.12</td>
<td>984</td>
</tr>
<tr>
<td>4.12</td>
<td>4.12.14</td>
<td>837</td>
</tr>
<tr>
<td>4.13</td>
<td>4.13.7</td>
<td>509</td>
</tr>
</tbody>
</table>

LTS EOLed LTS Stable
# of Yearly fixes in LTS

- LTS include 1 ~ 3 thousands of fixes every year
- Continue to apply these patches are very important for the security viewpoint

As of 2017/10/15

<table>
<thead>
<tr>
<th>Version</th>
<th>Maintainer</th>
<th>Released</th>
<th>Years maintained</th>
<th>Total Commits</th>
<th>Fixes/year</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.9</td>
<td>Greg Kroah-Hartman</td>
<td>2016-12-11</td>
<td>0.8</td>
<td>4038</td>
<td>4038</td>
</tr>
<tr>
<td>4.4</td>
<td>Greg Kroah-Hartman</td>
<td>2016-01-10</td>
<td>1.8</td>
<td>5619</td>
<td>3176.0</td>
</tr>
<tr>
<td>4.1</td>
<td>Sasha Levin</td>
<td>2015-06-21</td>
<td>2.3</td>
<td>4629</td>
<td>1989.3</td>
</tr>
<tr>
<td>3.16</td>
<td>Ben Hutchings</td>
<td>2014-08-03</td>
<td>3.2</td>
<td>7278</td>
<td>2266.2</td>
</tr>
<tr>
<td>3.10</td>
<td>Willy Tarreau</td>
<td>2013-06-30</td>
<td>4.3</td>
<td>6564</td>
<td>1523.8</td>
</tr>
<tr>
<td>3.2</td>
<td>Ben Hutchings</td>
<td>2012-01-04</td>
<td>5.8</td>
<td>8105</td>
<td>1397.5</td>
</tr>
</tbody>
</table>
LTSI Status
What is LTSI

• Open Source community to create and maintain LTSI kernel tree for long term
  – Based on LTS, All the LTS patches are applicable
  – Add another chance to include further patches on top of LTS, That is LTSI tree
  – Industry party to share best practice and help companies to use Linux for long term
LTSI includes LTS

**LTSI**
- Be able to add required features on top of LTS
- Share status, info, problem among industry people
- Huge testing by contributors
- Auto test frame-work
- Provide help developer for upstream

**LTS**
- Release 1 version / year, Maintain 2 years
- Frequently and large number of bug /security fixes
History of LTSI

• Established 2011, in ELCE Prague
  – 6 years now!
  – Started for stable Kernel for Android
    • Every Android release was used different kernel
    • Android 3.0 Ice cream Sandwich was used Linux 3.0
    • Number of different tree need to be integrated
  – Discussed about the shape of LTS
    • 2 Years term and release every year
    • Maintaining 2 LTS kernels is reasonable for both companies and maintainer
History of LTSI

• Maintained by Greg Kroah-Hartman, Fellow of Linux Foundation as same as LTS

• Released yearly basis; 3.0, 3.4, 3.10, 3.14, 4.1, 4.9

• Integrated by Yocto Project (2012, May)
  – Yocto is about 60% or more share of Embedded products

• Have had workshop/session to share information and discuss issue among industry

• Have many of use cases: AGL ···
Learned in 6 Years

The value of LTS/LTSI were:

1. LTS and LTSI is only a choice for the products
2. Upstream First policy
3. Security and Bug fixes are being more important in Embedded space
1. LTS and LTSI is only a choice

• For Long-term usage, LTS/LTSI is just fit
• LTS provides 2-3K of patches in a year
  – If the work should be done by a company, the company needs specific resources
• Now, all the Android device using LTS
  – LTS is default choice even for the other use case
• There is more longer term requirements
  – CIP, AGL and Android
2. Upstream First policy

• Changing kernel for the “product first” makes problem for long term use
  – Large number of fixes may NOT applicable in the future
  – Huge discussion happened before kernel summit 2016

• That’s why companies’ developers need participate Linux kernel community

• Initial hurdle may high but important for long term use
• LTSI is keeping upstream first policy
3. Security / Bug fixes are being more important

- Now fixing security problem is mandatory requirement
- To apply community provided security fixes, base code should be same as upstream. Otherwise Immediate patch release will not possible
  - In-house patches must as small as possible
Further steps for maintaining kernel long term
Case of Project Treble for Android

• Isolate Android OS and hardware specific code
  – Under the Vendor Specific Binder (/dev/vndbinder), all the vendor specific kernel code will run
  – VTS/CTS can test its interface
  – By this change, silicon specific patches and LTS patches can be applied separately
  – That makes Android software does NOT depend on Hardware
Live Patching

• Feature for live patching the kernel code was merged in Linux 4.0
  – Result of kgraft of SUSE and kpatch of RedHat
  – Most CVE can be safely to apply
  – X86 is primary architecture

• By using Live patching, some super important patch can be applied without down time
Kernel update mechanism

- CoreOS and ChromeOS have feature to update kernel.
  - There is 2 different partition. A is current working one and during working on A, new kernel will be downloaded into B and then boot from B.
  - Google is providing basic code called “Omaha”
  - Different commercial implementation is available
  - This will easier to upgrade kernel and also easy to roll back to previous kernel
Container based OS

• For Embedded space, Container will be able to used as packaging technology
  – Be able to ignore problem of Libraries and Language processor version

• Building Container OS is different
  – OS is just providing service for container
  – Basic OS should include minimum packages
  – Apps supports should be in Container
LTSI 2017 Development plan

Events:
- Announcements
- LTSI 4.9 Release
- Yocto2.4
- Electric Eel

Announce | Merge Window Open | Merge Window Close | Release
--- | --- | --- | ---
2/8 or 20 | End of June | End of July | End of August
You will be able to have chance to add new patches on top of 4.14 LTS in Merge Window next May.
Conclusion

• LTSI was started to fill the gap between community and industry but still there is the gap
  – We will continue our activity to discuss both side to better align each other

• Upstream first policy is important for Open Source

• Why don’t you join LTSI?
  – By joining LTSI, you will be able to share best practice
  – Be able to get information for stable kernel
THANK YOU
You can participate LTSI

- Follow on Twitter account: @LinuxLTSI
- Web: http://ltsi.linuxfoundation.org
- Mailing list: https://lists.linuxfoundation.org/mailman/listinfo/ltsi-dev
- Git tree: http://git.linuxfoundation.org/?p=ltsi-kernel.git;a=summary

LTSI stands for Long-Term Support Initiative. A group of CE Working Group of the Linux Foundation to provide Long-Term and stable Linux for Industry.