



LTSI Project update

Long Term Support Initiative

Tsugikazu SHIBATA, NEC

Hisao Munakata, Renessas

22, August 2014

LinuxCon North America@Chicago Sheraton

Contents



- Linux Development process
- LTSI Development process
- LTSI New challenges ; TEST Project

Status of Latest Linux Kernel

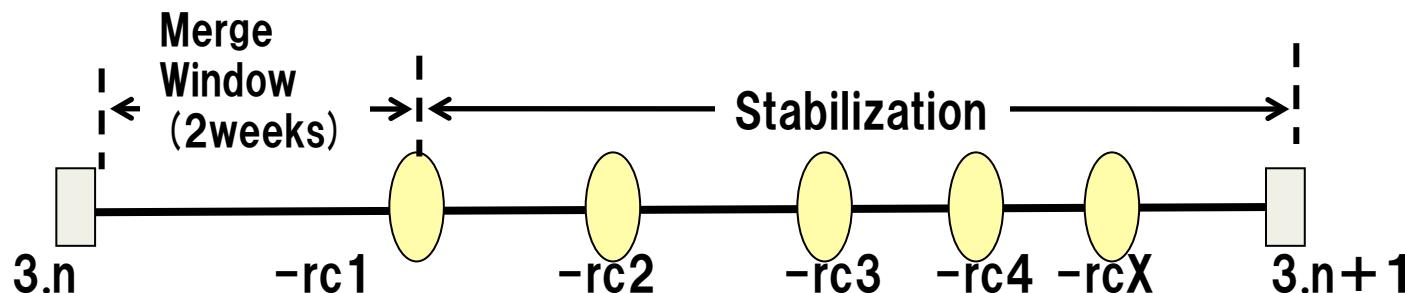


- Latest Kernel : 3.16
 - Released August 3rd
 - Lines of code : 18,881,498
 - Files : 47425
- Current Stable Kernels: 3.16.1
- Current development kernel: 3.17-rc1
 - Released: August 16

Linux Development process



- Just after the release of 3.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
- 3.n+1 will be released when it becomes reasonably stable by some of –rcX released



Kernel release cycle

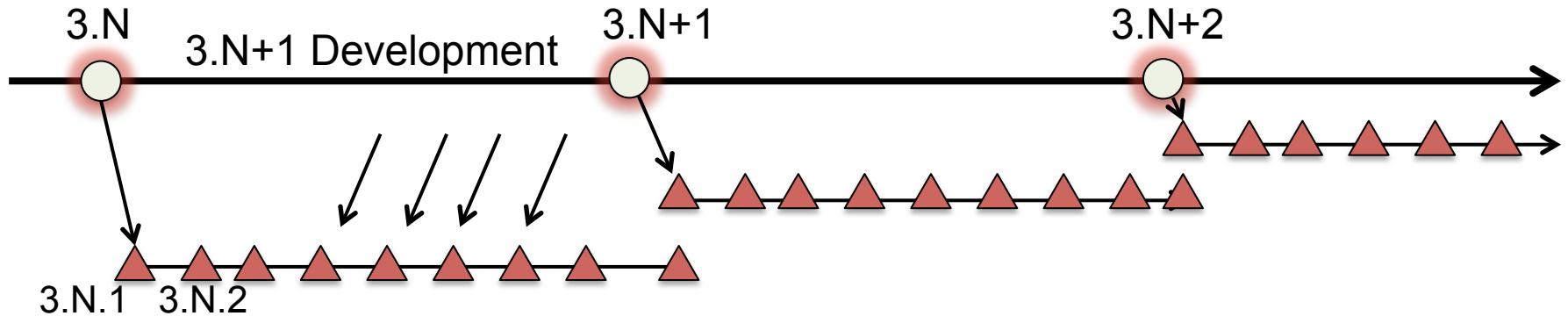


- Release cycle of Linux kernel is about 60-70 days

Version	Release	Duration
3.0	2011-7-21	
3.1	2011-10-24	95
3.2	2012-1-4	72
3.3	2012-3-18	74
3.4	2012-5-20	63
3.5	2012-7-21	62
3.6	2012-9-30	71
3.5	2012-12-10	71
3.8	2013-2-18	70
3.9	2013-4-28	69

Version	Release	Duration
3.10	2013-6-30	63
3.11	2013-9-2	64
3.12	2013-11-15	74
3.13	2014-1-21	67
3.14	2014-3-30	68
3.15	2014-6-8	70
3.16	2014-8-3	56

Stable kernel release



- 3 part version like 3.x.y
- Recommended branch for users who want the most recent stable kernel
- Contain small and critical fixes for security problems or significant regressions discovered in a latest development version
- Bug fixes/Security fixes are back ported from latest version
- Drop when next stable kernel were released

Stable_kernel_rules.txt

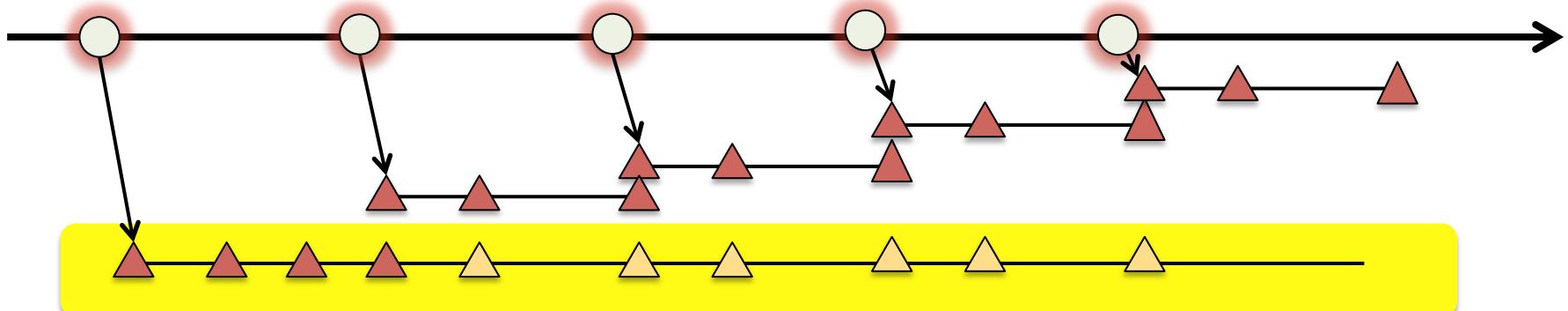
- Published in 2006. Strict rule to back port from latest version

See Documentation/stable_kernel_rules.txt

- It must be obviously correct and tested.
- It cannot be bigger than 100 lines, with context.
- It must fix only one thing.
- It must fix a real bug that bothers people.
- It must fix a problem that causes a build error, an oops, a hang, data corruption, a real security issue, In short, something critical.
- Serious issues as reported by a user may also be considered.
- New device IDs and quirks are also accepted.
- No "theoretical race condition" issues, unless an explanation of how the race can be exploited is also provided.
- It cannot contain any "trivial" fixes in it .
- It or an equivalent fix must already exist in Linus' tree (upstream).

LongTerm Stable Kernel

- Extended maintenance period for stable kernel
- Pick One version per year and maintain 2 years
- Bug and security fixes found in latest version are continued to back ported



LTS includes large number of fixes



- Stable release only back port important bug and security fixes
- Thousands of fixes ported

Version		# of commits
From	To	
3.0.1	3.0.101	3850
3.1.1	3.1.10	430
3.2.1	3.2.62	5172
3.3.1	3.3.8	522
3.4.1	3.4.103	4624
3.5.1	3.5.7	694
3.6.1	3.6.11	700
3.7.1	3.7.10	690
3.8.1	3.8.13	841
3.9.1	3.9.11	631

Version		# of commits
From	To	
3.10.1	3.10.53	3302
3.11.1	3.11.10	660
3.12.1	3.12.26	2742
3.13.1	3.13.11	892
3.14.1	3.14.17	1346
3.15.1	3.15.10	690

LTS includes large number of fixes



- Stable release only back port important bug and security fixes
- Thousands of fixes ported

Version		# of commits
From	To	
3.0	3.0.101	3953
3.1	3.1.10	695
3.2	3.2.62	5222
3.3	3.3.8	698
3.4	3.4.103	4713
3.5	3.5.7	816
3.6	3.6.11	757
3.7	3.7.10	718
3.8	3.8.13	996
3.9	3.9.11	746

Version		# of commits
From	To	
3.10	3.10.53	3322
3.11	3.11.10	677
3.12	3.12.26	2762
3.13	3.13.11	903
3.14	3.14.17	1370
3.15	3.15.10	703
3.16	3.16.1	18

Current LTS versions



Version	Maintainer	Released	Projected EOL
3.14	Greg Kroah-Hartman	2014-3-30	Aug 2016
3.12	Jiri Slaby	2013-11-3	2016
3.10	Greg Kroah-Hartman	2013-6-30	Sep, 2015
3.4	Greg Kroah-Hartman	2012-5-20	Oct 2014
3.2	Ben Huchtings	2012-1-4	2016
2.6.32	Willy Tarreau	2009-12-3	Mid 2015

<https://kernel.googlesource.com/pub/scm/docs/kernel/website/>

Key activities of LTSI

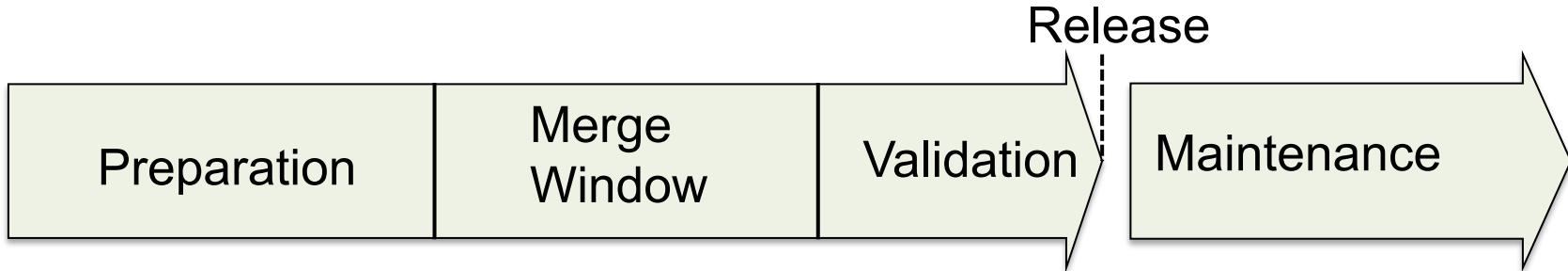
Long Term Support Initiative



- Provide a industry managed kernel and maintain Long term based on LTS
- Provide a common place for embedded industry to share information: Mailing list, Workshop ...
- Provide place to support upstream activity : Help industry developer



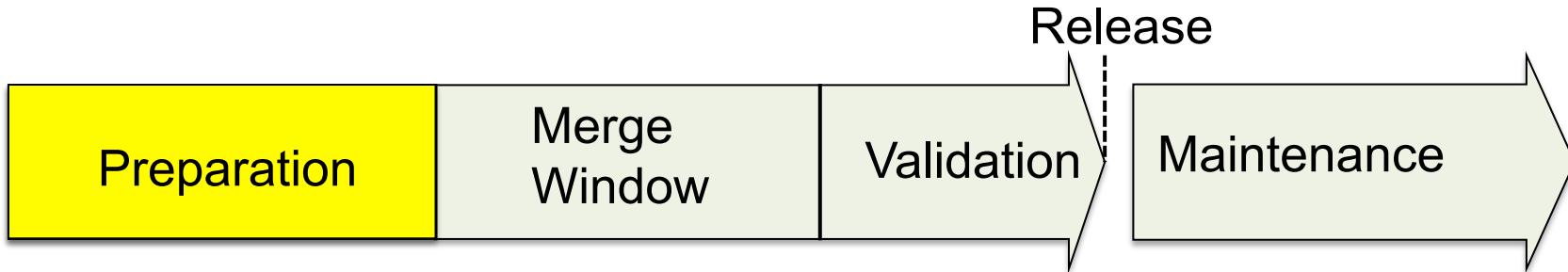
LTSI development process



Discussing about process as;

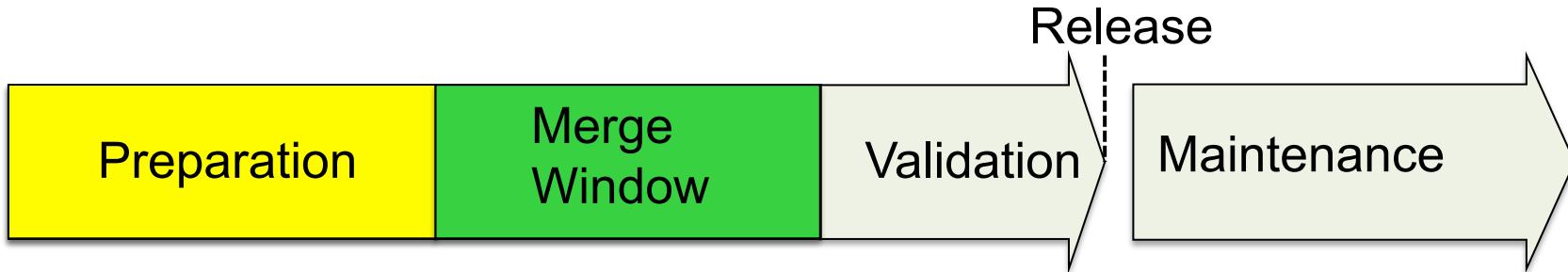
- Preparation
- Merge window
- Validation
- Maintenance

LTSI development process



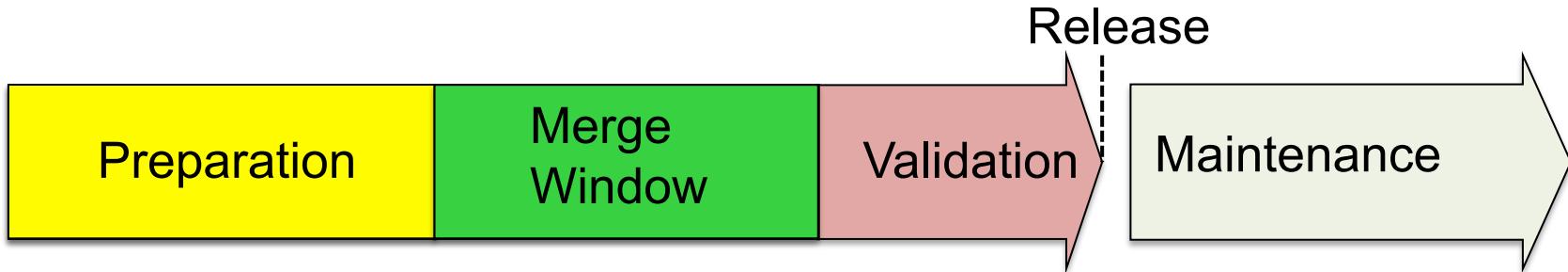
- Decide LTS version
- Decide LTSI development schedule
- Each developer should;
 - Merge own patches into upstream to prepare merge into LTSI. That will be easier to maintain production kernel (ie. Driver)
 - Prepare to back port important features in the upstream to LTSI

LTSI Process: Merge window



- Merge window will open after 4-5 month of LTS release
- Anyone can propose patches in the merge window period
- Patches should be in the upstream or at least in the –next
- Exceptional case: beneficial for the wide range of users, or projected to be in the upstream
- Merge window period is about two month and –rc1 will be released right after the window was closed
- No additional patches are accepted after the window closed

LTSI process: Validation



- After release of –rc1, developer should validate whether its features were correctly merged and worked fine
- Patches to fix problem should be sent
- Validation period is about a month or more
- LTSI development will be finished when maintainer recognized reasonably stable and then new LTSI kernel will be released

Patchwork is ready

- Patchwork is tool to cherry pick patches posted on ML
- LTSI providing patchwork
- In case you still have patches after closed the merge window, you can send patch to LTSI ML anytime.
- Using Patchwork, you can share patches even after the release

Patchwork LTSI Project development

[login](#) [register](#) [mail settings](#)

[Project: Itsi-dev : patches : project info : other projects](#) [about](#)

Incoming patches

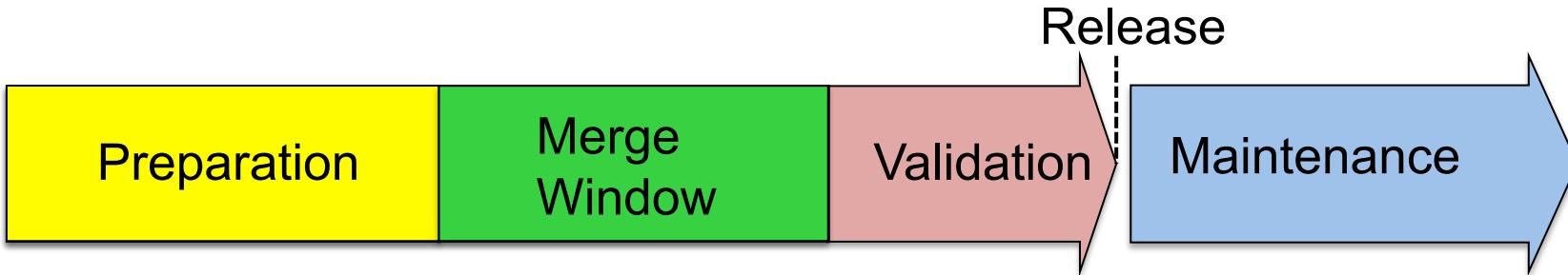
[« Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [Next »](#)

Filters: State = Action Required  | Archived = No 

Patch	Date	Submitter	Delegate	State
[12/12] ARM: shmobile: armadillo800eva: set clock rates before timer init	2013-05-22	Do Quang Thang		New
[11/12] ARM: shmobile: use FSI driver's armadillo800eva				
[10/12] ASoC: fsl_tidiumpch_fsi_platform_info pointer	2013-05-22	Do Quang Thang		New

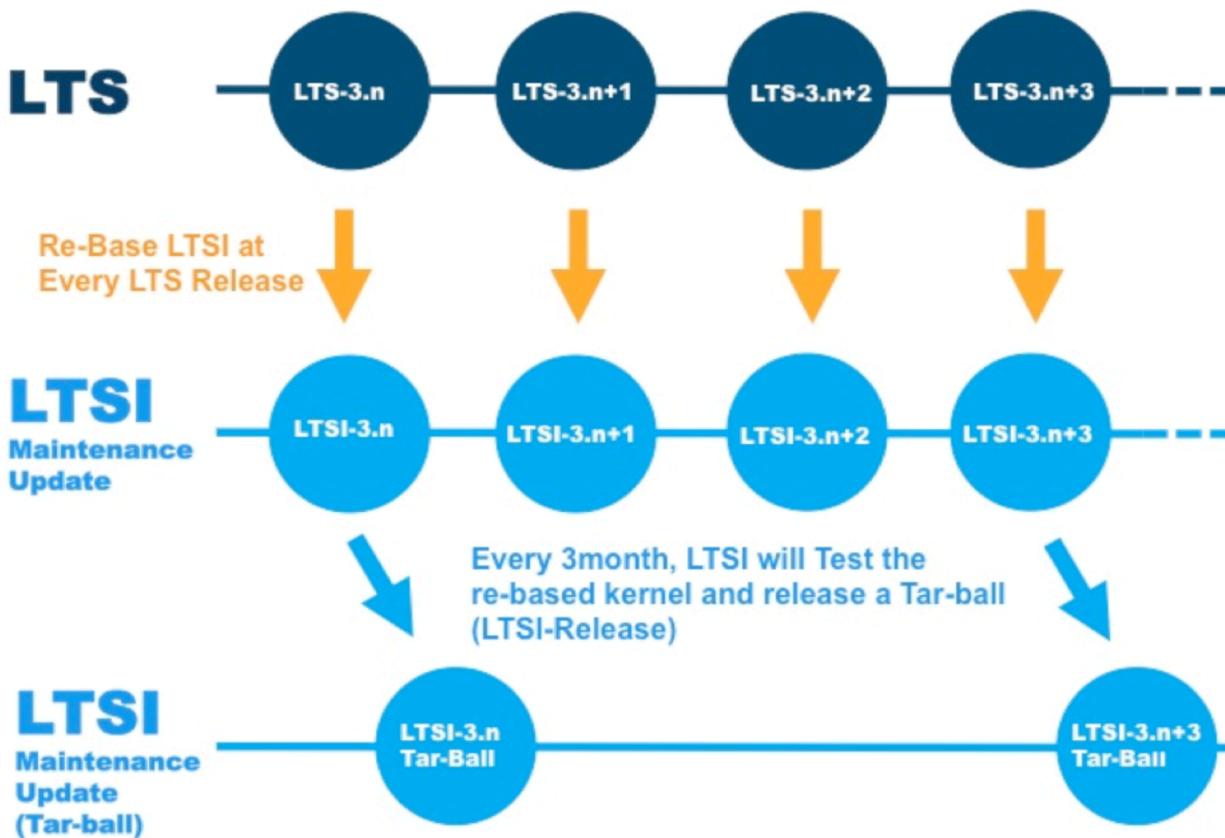
<https://patchwork.kernel.org/project/ltci-dev/list/>

LTST Process: Maintenance



- LTS is continue to maintain in two years since that version was released
- LTSI will be rebased at the same time each new LTS was released
 - If you want to create own kernel based on LTS such as 3.10.x, you will be able to find same version in LTSI git tree
 - There are tool to create LTSI full kernel source in scripts directory
- LTSI full source tarball will be released every 3month after QA team have done its test as a volunteer work

LTSI: maintenance stream



LTSI-3.10 history

Event	Date
kernel 3.10 merge window open	2013.4.28
kernel 3.10 merge window close	2013.5.12
kernel 3.10 release	2013.6.30
Announce of 3.10 becomes LTS	2013.8.4
LTSI-3.10 merge window open	2013.11.13
Merge Window period	75 days
LTSI-3.10-rc1 (=merge window close)	2014.1.29
Validation period	26 days
LTSI-3.10 release	2014.2.24

Major contributors for LTSI-3.10



Contributor	Patch count
Darren Hart (Intel)	1,197
Simon Horman (for Renesas)	1,122
Daniel Sangorrin (Toshiba)	123
Patrik Jakobsson (for Intel)	46
Mark Brown (linaro)	11
Greg Kroah-Hartman (Linux Foundation)	11
Total	2,510

Major achievement of LTSI-3.10



- LTTng
- Power efficient work queues
- Intel BayTrail support
- Intel Minnow board support
- Renesas R-Car H2/M2 series support
- Xilinx Zinq board support

NEW

LTSI-3.14 Release plan



Event	Date
kernel 3.14 merge window open	2014.1.19
kernel 3.14 merge window close	2014.2.2
kernel 3.15 release	2014.3.30
3.14 becomes LTS	2014.7.3
LTSI-3.14 merge window open	2014.8.21
Merge window period	
75 days	
LTSI-3.14-rc1 (=merge window close)	2014.10.30?
Validation period	
50+ days	
LTSI-3.14 release	2014.12.25?

Please send patch to LTSI mailing list now!

LTSI NEW CHALLENGES

TEST PROJECT

LTSI test : Background



- LTSI is patchset add on top of LTS
- LTS is quite stable because of picking patches with strict rule (stable_kernel_rules.txt)
- However, LTSI will include
 - Back port features in latest upstream
 - Industry demanded non mainlined code

We want to validate LTSI kernel that does not include any bug or regression against LTS

LTSI test : Share test case!



- Now LTS and LTSI is used by lots of companies
- Each companies may spend a lot of time for its validation
- Some of fundamental feature test might be duplicated and such portion might be shared

We can consider sharing some part of kernel test across the industry

- **Test environment**
- **Test case**

LTSI test environment



- We have developed LTSI test environment
 - Fully automated execution
 - Easy to manage
 - Monitoring what have happened
 - GUI
 - Deliverable (Be able to install each companies)
 - 28 benchmarks and 33 test programs are integrated in initial test environment

LTSI test environment



Current stage is just started but You can down load from
<https://bitbucket.org/cogentembedded/jta-public/>

Atlassian Bitbucket Features Pricing owner/repository ? English Sign up Log in

cogentembedded jta-public

ACTIONS Clone Compare Fork

NAVIGATION Overview Source Commits Branches Pull requests Issues Wiki Downloads

Overview

Last updated 2014-07-31 Language — Access level Read

9 Branches	0 Tags
0 Forks	1 Watcher

HTTPS → https://bitbucket.org/cogentembedd

Unlimited private and public hosted repositories. Free for small teams!

Sign up for free

Jenkins-based Test Automation

8. Software requirements

Debian Wheezy 64-bit.

1. Installing

run ./debian_install.sh under root, answer simple questions.

It will install all the needed packages.

Install toolchains for needed platforms to /home/jenkins/tools and add them to tools.

See jta-guide.pdf for details.

2. Usage

Recent activity

2 commits Pushed to cogentembedded/jta-public 7cefe53 jobs: FE: add renesas jobs for te... 38e2fef tests: add renesas tests (backend) 4da · 2014-08-01

1 commit Pushed to cogentembedded/jta-public 828393b tests: exit 1 -> build error for buil... 4da · 2014-07-26

1 commit Pushed to cogentembedded/jta-public 1074c68 scripts: add build_failed function ...

<https://bitbucket.org/cogentembedded/jta-public/>

LTSI test environment



- Initial documentation:

<https://bytebucket.org/cogentembedded/jta-public/raw/7cefe53a09b5028bf2c99663d81ecde39b486713/docs/jta-guide.pdf>

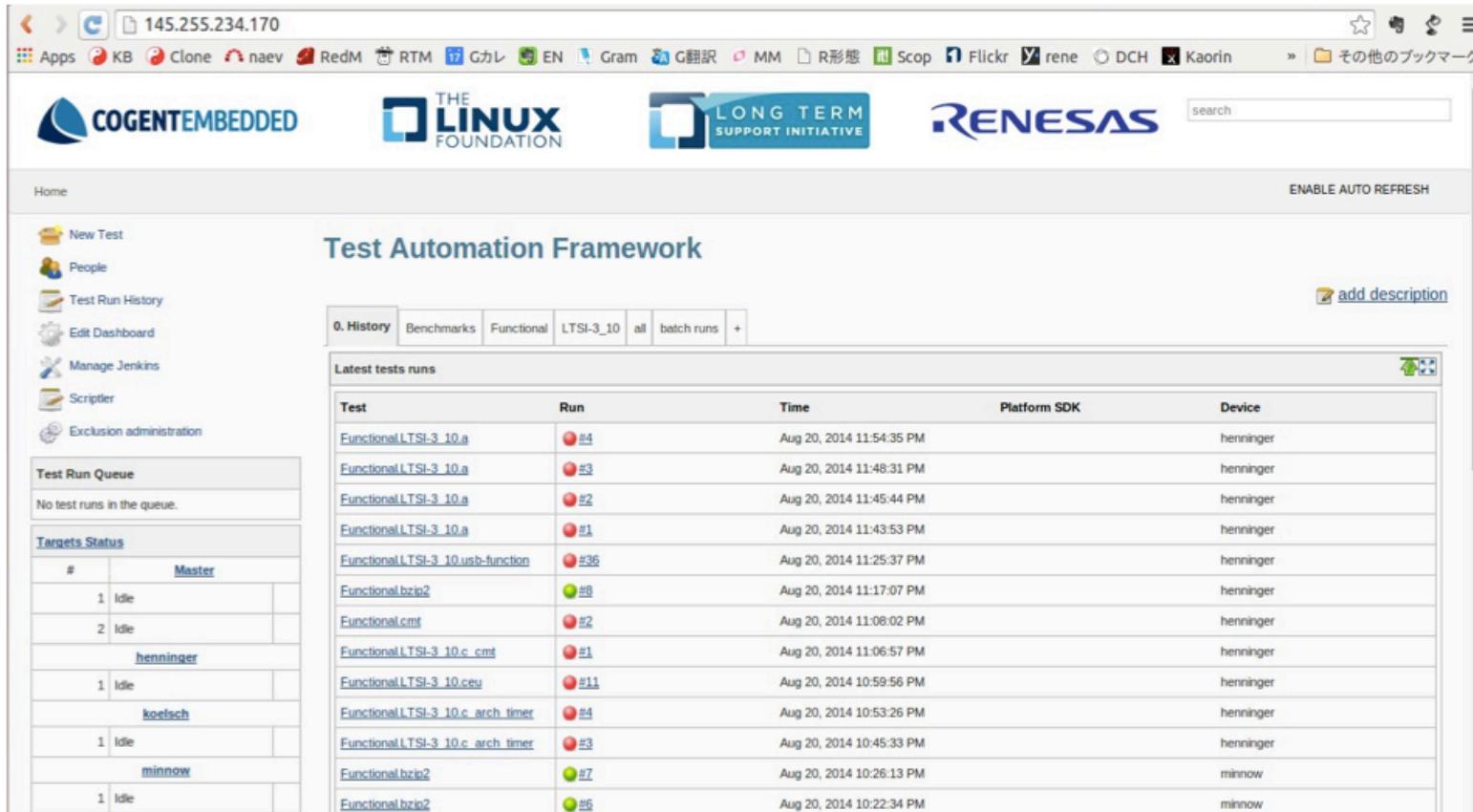
Your feed back is welcome!

NEXT STEP for LTSI test environment



- Daily/Nightly test of LTSI for specific hardware
 - Public server 24h/7d up/running for LTSI testing with Intel Minnow, Renesas Henninger
- More I/O and platform-specific tests
- Polished docs, deployment/installation scripts

You will be able to refer its result



New Test

People

Test Run History

Edit Dashboard

Manage Jenkins

Scripter

Exclusion administration

Test Run Queue

No test runs in the queue.

Targets Status

#	Master
1	Idle
2	Idle
	henninger
1	Idle
	koelsch
1	Idle
	minnow
1	Idle

Test Automation Framework

0. History Benchmarks Functional LTSI-3_10_a all batch runs + add description

Test	Run	Time	Platform SDK	Device
Functional LTSI-3_10_a	②4	Aug 20, 2014 11:54:35 PM		henninger
Functional LTSI-3_10_a	②3	Aug 20, 2014 11:48:31 PM		henninger
Functional LTSI-3_10_a	②2	Aug 20, 2014 11:45:44 PM		henninger
Functional LTSI-3_10_a	②1	Aug 20, 2014 11:43:53 PM		henninger
Functional LTSI-3_10_usb-function	②36	Aug 20, 2014 11:25:37 PM		henninger
Functional bztp2	②8	Aug 20, 2014 11:17:07 PM		henninger
Functional cmt	②2	Aug 20, 2014 11:08:02 PM		henninger
Functional LTSI-3_10_c_cmt	②1	Aug 20, 2014 11:06:57 PM		henninger
Functional LTSI-3_10_ceu	②11	Aug 20, 2014 10:59:56 PM		henninger
Functional LTSI-3_10_c_arch_timer	②4	Aug 20, 2014 10:53:26 PM		henninger
Functional LTSI-3_10_c_arch_timer	②3	Aug 20, 2014 10:45:33 PM		henninger
Functional bztp2	②7	Aug 20, 2014 10:26:13 PM		minnow
Functional bztp2	②6	Aug 20, 2014 10:22:34 PM		minnow

<http://145.255.234.170/>

Test case sharing



- We have discussed at LTSI workshop
- What we have discussed :
 - License to deliver; strong or weak?
 - How the test cases are maintained; maintainer or just gather
 - How we manage test cases; collect test case and referencing test cases

Further discussion will be happen in our workshop, please join us!

THANK YOU



You can participate LTSI

- Follow on Twitter account:

@LinuxLTSI



LinuxLTSI

@LinuxLTSI

LTSI stands for Long-Term Support Initiative. A group of CEOs from leading Linux companies that work together to provide Long-Term and stable Linux for Industry.

- Web:

<http://ltsi.linuxfoundation.org>

- Mailing list:

<https://lists.linuxfoundation.org/mailman/listinfo/ltsi-dev>

- Git tree :

<http://git.linuxfoundation.org/?p=ltsi-ernel.git;a=summary>