

Review of the Stable Realtime Release Process

An unscientific, slightly opinionated stab at the current status...

With the intent of generating some discussion.

Where this info comes from

My experiences

Scanning linux-rt-users email list

Request for feedback on linux-rt-users email list

Scanning linux-rt-users email list

Culled from emails that I had not bothered to delete from my mailbox. So there probably were other relevant emails that I have deleted.

41 people

72 messages

Stable Series First Version

3.0.10-rt27

3.2.15-rt25

3.4.1-rt9

Versions Referenced

2.6.33.9-rt31

2.6.33.20-rt31

3.0.9-rt25

3.0.10-rt27

3.0.10-rt27 Ubuntu 10.04.3 LTS

3.0.12-rt29

3.0.12-rt30

3.0.12-rt30-rc1

3.0.12-rt30-rc2

3.0.12-rt30-rc3

3.0.14-1-rt31

3.0.14-1-rt32-rc1

3.0.14-rt31

3.0.14-rt32

3.0.20-rt35

3.0.20-rt36

3.0.23-rt29

3.0.25-rt44

3.0.26-rt45

3.0.30-rt50

3.0.31-rt51

3.0.34-rt55

3.0.36-rt57

3.0.36-rt58

3.0.36-rt58-rc1

Versions Referenced

3.2-rc4-rt6	3.4.2-rt10	
3.2.9-rt17	3.4.3-rt11	
3.2.12-rt22	3.4.3-rt11	under xen
3.2.16-rt27	3.4.4-rt13	
3.2.17-rt28	3.4.8-rt16	(fedora 17)
3.2.17-rt28-fc16	3.4.9-rt17	
3.2.18-rt29		
3.2.20-rt32		
3.2.23-rt36		
3.2.23-rt37		
3.2.27-rt40		

Systems Referenced

ARM dual-core Cortex-A9	HP Compaq 6710s laptop
ARM i.mx pcm043	i686 core2 6300
ARM i.mx35	i686, x86_64
ARM Pandaboard	intel 24 core
ARM PCM043	mips Toshiba TX4938
ARM RealView	powerpc MPC85xx
ARM Naviengine	powerpc MPC8xx custom
Dual Opteron 244	ppc64, Freescale P5020
Intel I7 860	ppc64, MacPro G5
SMP AMD64	raspberry pi
HP Compaq 6710s laptop	thinkpad t410, x86_64

Email Content

- fix
- Osadl Stable vs Rostedt stable question
- performance regression
- problem
- reproduce problem
- test

Request for feedback

- 1) Have you downloaded any stable RT releases? If so, approximately how many? If not, why not?
- 2) How do you use the stable RT release? To run on a single system? To place on a product that is sold / lent / given away? To create a distribution? Other?
- 3) Is the stable RT release useful and valuable? (How many beers do we owe to Steve when we run into him at conferences?) Why is it useful and valuable?
- 4) If the stable RT release is not useful, why isn't it?
- 5) What changes would make the stable RT release more useful? Why would the changes make it more useful?

Request for feedback

- 6) What changes would make the stable RT release better?
- 7) What should `_not_` change about the stable RT release?
- 8) How frequently do you test a stable RT release (percent of releases)?
- 9) How intensively do you test a stable RT release? Boot? Functional test? Performance test(s)? Other test(s)? Single target (type of computer)? Multiple targets? How many different types of targets?
- 10) Which series have been useful? Which series is currently most useful (can be more than one)? (3.0.x, 3.2.x, 3.4.x)
- 11) Any other random or useful comments or opinions?

Request for feedback

7 responses

Request for feedback

1) Have you downloaded any stable RT releases? If so, approximately how many?
If not, why not?

- All
- almost every update on the stable branch I'm tracking.
- All of them.
- 3.2, which was the latest stable release available when we needed one.
- about every second release.
- Yes, 24.

Request for feedback

2) How do you use the stable RT release? To run on a single system? To place on a product that is sold / lent / given away? To create a distribution? Other?

- build and test enterprise kernels.
- control our industrial machines, which are sold all around the world.
- runs on 20-30 developer computers
- We run a test center for quality assessment ...
- From time to time, we release a "Latest Stable" real-time kernel version

Request for feedback

2) How do you use the stable RT release? To run on a single system? To place on a product that is sold / lent / given away? To create a distribution? Other?

- intend to sell a product that contains a stable RT kernel
- a product that is currently under development and is about to be released for sale at the end of this year
- internal distro, will be used to create consumer products

Request for feedback

3) Is the stable RT release useful and valuable? (How many beers do we owe to Steve when we run into him at conferences?) Why is it useful and valuable?

- [beers] Steven, Thomas, and all their friends, couldn't possibly put a dent in it and remain conscious ;-)
- It allows to stay close to the distributions kernel version.
- It gives some stability.
- Creates a better quality base to build on.

Request for feedback

3) Is the stable RT release useful and valuable? (How many beers do we owe to Steve when we run into him at conferences?) Why is it useful and valuable?

- The product we intend to sell is marketed for its determinism.
- we've used Steven's rebased versions ... which we've found useful to maintain a clean history
- having one developer maintaining a public stable rt release avoids a lot of the same work being done by lots of people

Request for feedback

4) If the stable RT release is not useful, why isn't it?

- Because it is not stable. Should it ever become stable, nobody would notice.

Request for feedback

5) What changes would make the stable RT release more useful? Why would the changes make it more useful?

- Softirqs dying would be great, they're a thread meets lock rendezvous
- We would need to establish a transparent release procedure and a list of known bugs

Request for feedback

5) What changes would make the stable RT release more useful? Why would the changes make it more useful?

It would be useful to know what systems a given release have been run on (e.g., OSADL).

Is there prerelease criteria for a stable RT release?
If so, what are they?

Mainly interested in what processors and chipsets have been tested and test results (e.g., cyclicttest).
If something like this existed, we would be willing to submit results."

Request for feedback

6) What changes would make the stable RT release better?

- Fix bugs, before you release
- Maintain a list of undefined, unresolved, unfixed etc. bugs [Bugzilla suggested]
- every report to the linux-rt-user mailing list should go to this Bugzilla system
- Fewer concurrent branches, to concentrate testing, fixes. Downside could be that the branch we care about is dropped.

Request for feedback

7) What should `_not_` change about the stable RT release?

- That they're maintained and released should definitely `_not_` change.
- I really like that Steve put this thing into a git repository. It makes tracking vanilla RT much simpler.

Request for feedback

8) How frequently do you test a stable RT release (percent of releases)?

- 100% of 3.0, others not so frequently, days being made of finite hours
- I usually test major updates to a new kernel version or if I change the kernel configuration. But I do some trivial testing for every new kernel before I distribute them to my fellow developers.
- 100%

Request for feedback

8) How frequently do you test a stable RT release (percent of releases)?

- Since we started, we have tested two stable 3.2 RT releases
- I test about 50% of stable releases
- 24. Some get more extensive testing (the ones that go into the internal development tree).

Request for feedback

9) How intensively do you test a stable RT release? Boot? Functional test? Performance test(s)? Other test(s)? Single target (type of computer)? Multiple targets? How many different types of targets?

- All of the above, but for not enough targets, not enough tests and not enough hours. Primary targets are small IBM, and largish HP.
- We perform our day-to-day linux development on a stable-rt kernel.
- For every stable update that I pull in I usually run cyclicttest for a few hours on my desktop and do some browsing, compilation etc.

Request for feedback

9) How intensively do you test a stable RT release? Boot? Functional test? Performance test(s)? Other test(s)? Single target (type of computer)? Multiple targets? How many different types of targets?

- For new releases or kernel configuration changes I deploy the kernel to one of our machine control PCs and have it drive an EtherCAT Bus for 2-3 days. The software sends a new frame to the bus every 250us. If the time between 2 frames exceeds 2ms, the hardware will cause a bus error, which tells me that something is wrong.
- Our computers run various intel CPUs with a 64b kernel

Request for feedback

9) How intensively do you test a stable RT release? Boot? Functional test? Performance test(s)? Other test(s)? Single target (type of computer)? Multiple targets? How many different types of targets?

- We run 100 million cyclicttest wakeup cycles twice a day (takes 5 hours and 33 minutes each).
- In addition, we test for real-time performance under idle, moderate and heavy CPU load and under accelerated graphics load. We also check for performance and latency regressions and for stability. About 80 different boards with four main architectures are continuously under test.

Overview: <https://www.osadl.org/?id=879>.

Continuous latency monitoring: <https://www.osadl.org/?id=864>

Latency plots: <https://www.osadl.org/?id=1313>

Request for feedback

9) How intensively do you test a stable RT release? Boot? Functional test? Performance test(s)? Other test(s)? Single target (type of computer)? Multiple targets? How many different types of targets?

- We run system-level tests on top of the RT release. We test interrupt latency, jitter (baseline, with malloc, with peripheral drivers), file I/O throughput, network throughput, floating point performance, boot time. We also test for regression of available memory or disk.
- Tests are functional tests on a single target (Freescale i.MX31 based device).

Request for feedback

9) How intensively do you test a stable RT release? Boot? Functional test? Performance test(s)? Other test(s)? Single target (type of computer)? Multiple targets? How many different types of targets?

- Light testing (18 versions):

 - 3 ARM targets

 - Compile and boot

- Heavy testing (6 versions):

 - 4 ARM targets

 - Large variety of kernel configurations.

 - Large number of functional tests.

 - Performance tests. Throughput, latencies, scheduler related.

Request for feedback

10) Which series have been useful? Which series is currently most useful (can be more than one)? (3.0.x, 3.2.x, 3.4.x)

- 2.6.x and 3.0.x constitute the main course on my dinner plate, so those.
- 2.6.31 for a very long time, then 3.0 for some time. Since april we run 3.2.23-rt37 on our development computers, the machine computers are still on 3.2.17-rt28.
- no series can be recommended for production yet

Request for feedback

10) Which series have been useful? Which series is currently most useful (can be more than one)? (3.0.x, 3.2.x, 3.4.x)

- 3.2.x
- Currently we use the 3.0.x series but are considering to switch to the 3.4.x series because of its newly announced long-term support
- 3.0.x

Request for feedback

11) Any other random or useful comments or opinions?

No additional comments

Request for feedback

[One response appeared to be about 2.6.x]

I have used rt kernel in industrial control, however not 'stable'

In general the latest kernel available, have been picked, and from there on stay with latest since all developement is on latest, when you got something stable, pick that kernel and stay with it for that release of hard and software.

Request for feedback

These were embedded controllers, running a closed loop, so we did test and got statistics from hardware measurements, (a bit of timing and such in external hardware), but also had watchdogs in hardware so to much jitter or overrun 'tripped' the system.

So stay with latest until youre done, then stop at stable were our strategy.

Random Thoughts

What trees (does Steve) / (should Steve) want to maintain?

Greg KH is currently maintaining stable kernel trees for the following amount of time:

- 3.0 - for at least one more year

- 3.4 - for at least two years

- 3.5 - until 3.6.1 is out

<http://lkml.indiana.edu/hypermail/linux/kernel/1208.2/02624.html>

Ben Hutchings is maintaining: 3.2.17 ++

<http://lkml.indiana.edu/hypermail/linux/kernel/1204.2/03657.html>

Random Thoughts

What trees (does Steve) / (should Steve) want to maintain?

Answer - intent is to:

- Follow Greg KH stable trees
- Currently following Ben Hutchings Debian tree

Random Thoughts

git vs quilt ---- please, no religious arguments

- Steve's choice, he is doing the work
- Some of the feedback commented on the value of git
- The release and -rebase tags make it possible to use 'git format-patch' to create a set of broken out files for the quilt model
- Steve initially warned that the -rebase tags may not be available always

Random Thoughts

git vs quilt ---- please, no religious arguments

- I depend upon the rebasing, so I can create a broken out series for the quilt model
- Steve may be considering creating the broken out series as one of his release delivery methods

Random Thoughts

There is recurring confusion due the to overloading of “stable”.

- Steve's releases are following the naming convention of the Greg KH stable releases.
- OSADL has a history of labeling releases as “OSADL latest stable”.

Random Thoughts

How to reduce confusion about what is “stable”?

- Frank will document the two terms on rtwiki.
- Should Steve's release announcements have boilerplate to explain the difference, or say it is not the “OSADL latest stable”?
- Should the OSADL latest stable web page explain the difference?

Random Thoughts

- Are the stable RT releases a big workload for Steve?
- Any other things to discuss?

