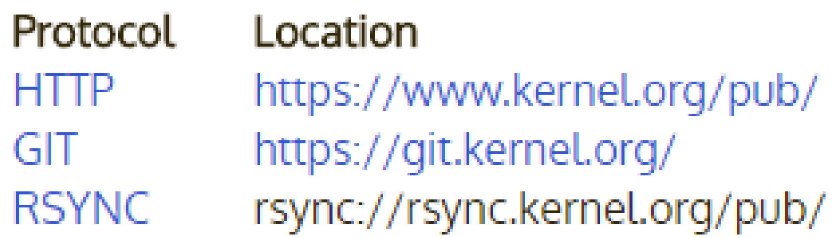
LONG TERM SUPPORT INITIATIVE

LTSI 2016 review

2nd Dec 2016 : CEWG Jamboree #59

Hisao Munakata

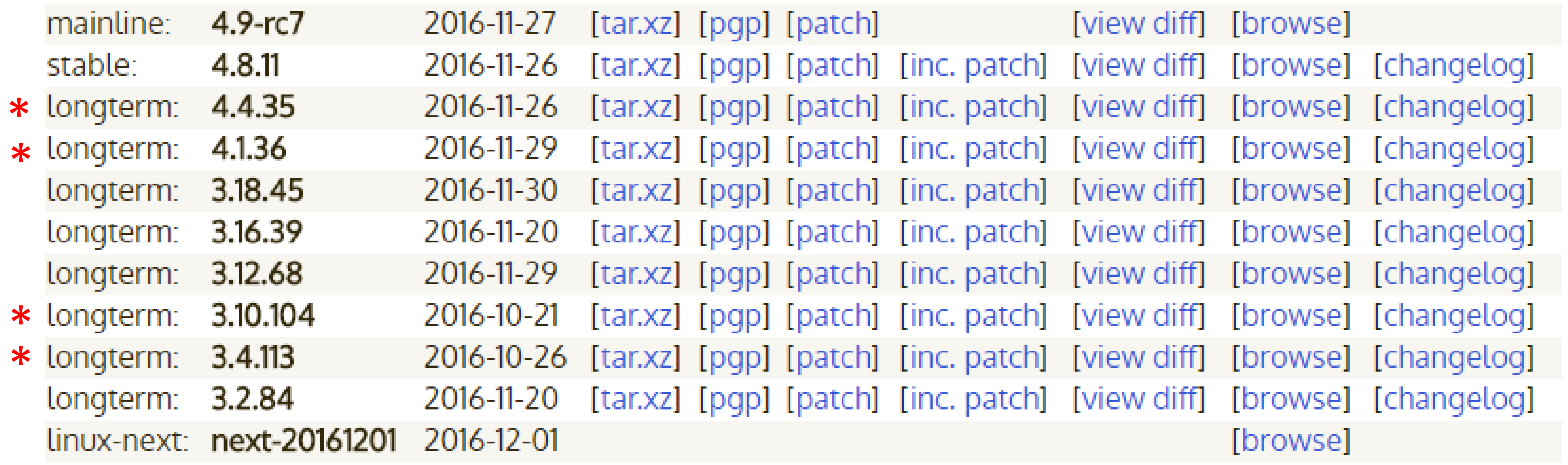
upstream kernel



Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

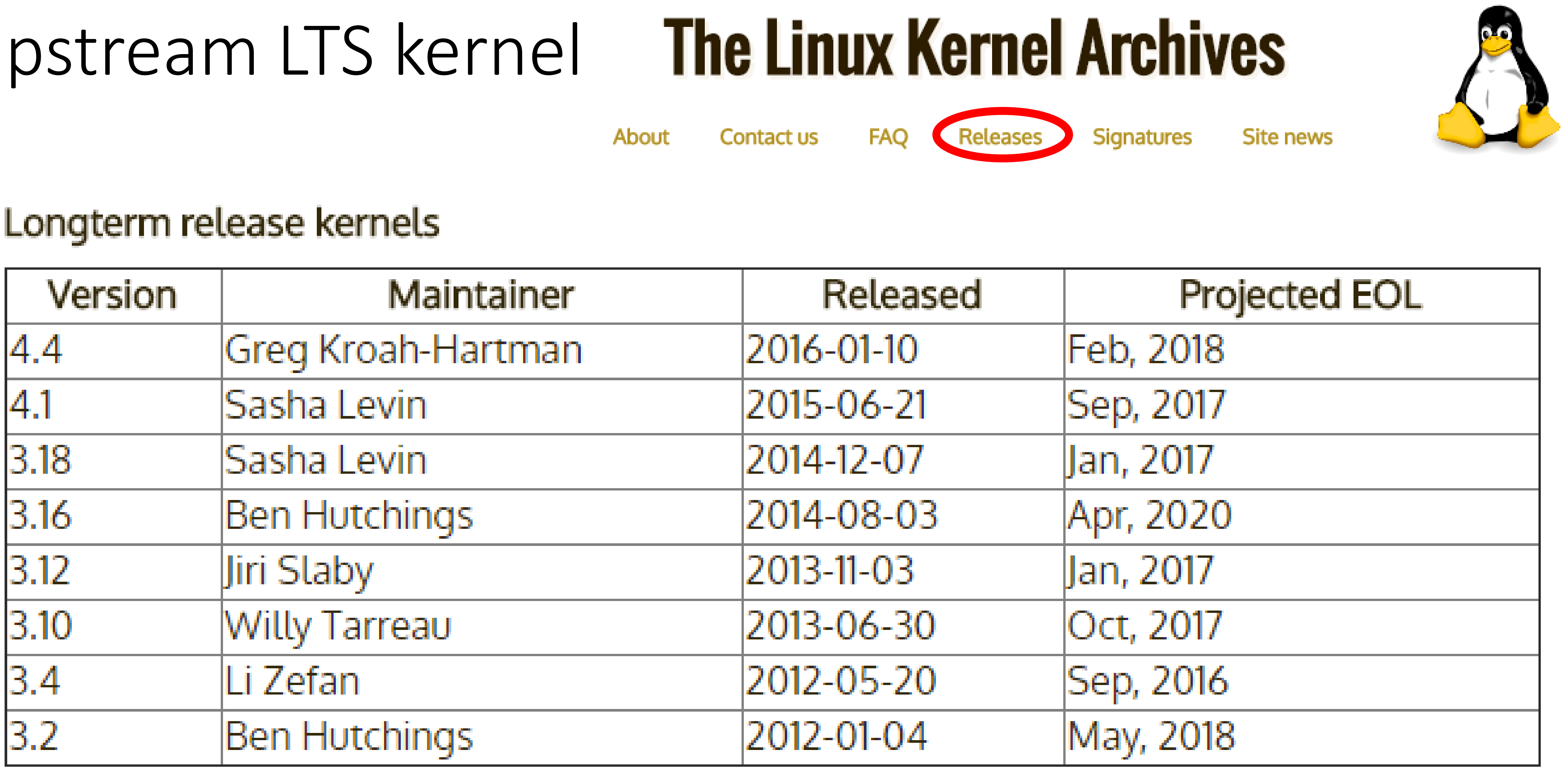
Latest Stable Kernel:

4.8.11



	mainline:	4.9-rc7	2016-11-27	[tar.xz]	[pgp]	[patch]		[view diff]	[browse]	
	stable:	4.8.11	2016-11-26	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
*	longterm:	4.4.35	2016-11-26	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
*	longterm:	4.1.36	2016-11-29	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
	longterm:	3.18.45	2016-11-30	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
	longterm:	3.16.39	2016-11-20	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
	longterm:	3.12.68	2016-11-29	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
*	longterm:	3.10.104	2016-10-21	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
*	longterm:	3.4.113	2016-10-26	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
	longterm:	3.2.84	2016-11-20	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
	linux-next:	next-20161201	2016-12-01						[browse]	

upstream LTS kernel



The Linux Kernel Archives

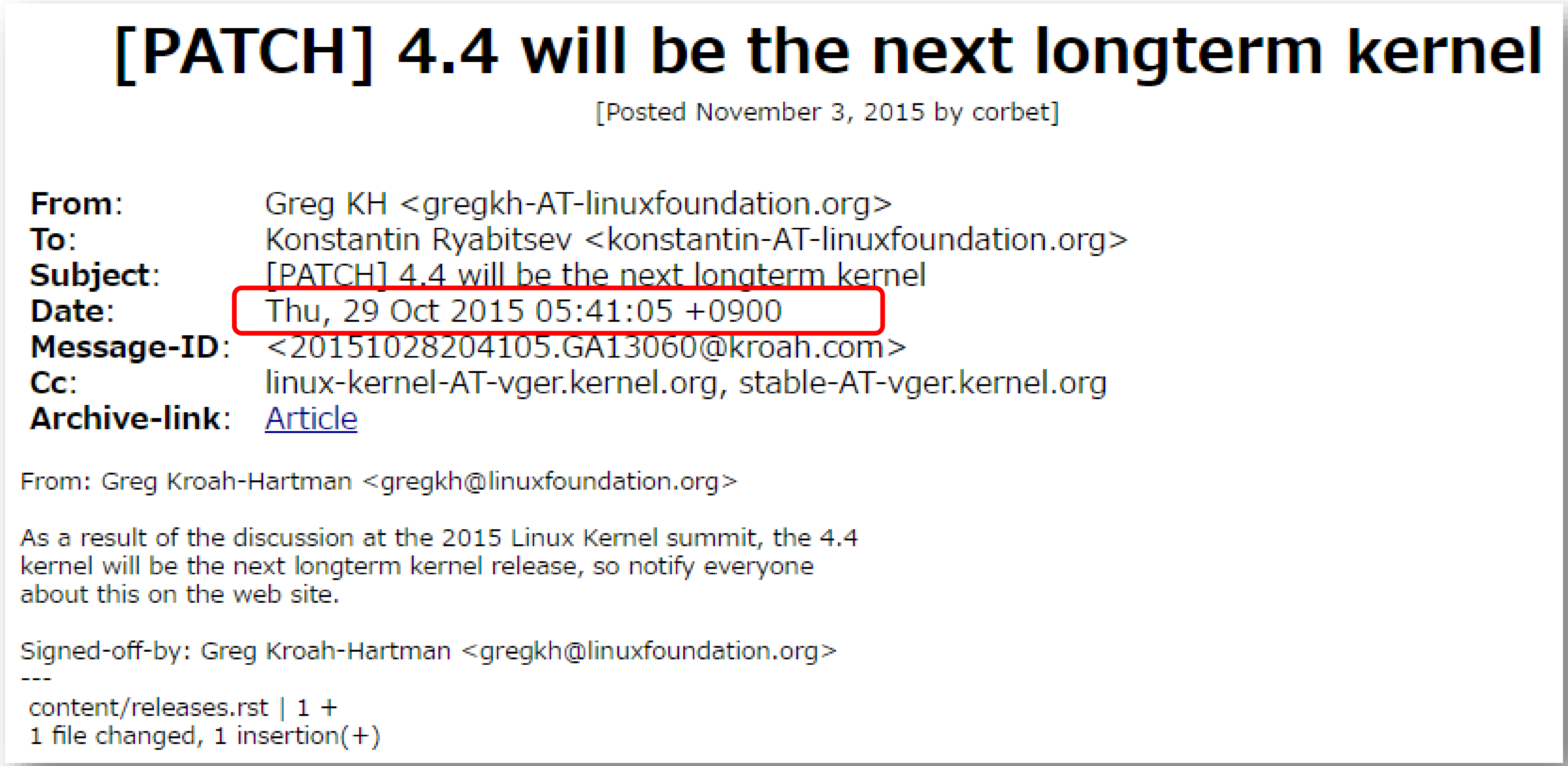
About Contact us FAQ Releases Signatures Site news

Longterm release kernels

	Version	Maintainer	Released	Projected EOL
*	4.4	Greg Kroah-Hartman	2016-01-10	Feb, 2018
*	4.1	Sasha Levin	2015-06-21	Sep, 2017
	3.18	Sasha Levin	2014-12-07	Jan, 2017
	3.16	Ben Hutchings	2014-08-03	Apr, 2020
	3.12	Jiri Slaby	2013-11-03	Jan, 2017
*	3.10	Willy Tarreau	2013-06-30	Oct, 2017
*	3.4	Li Zefan	2012-05-20	Sep, 2016
	3.2	Ben Hutchings	2012-01-04	May, 2018

https://www.kernel.org/category/releases.html

LTS 4.4 decision at kernel summit 2015



[PATCH] 4.4 will be the next longterm kernel

[Posted November 3, 2015 by corbet]

From:	Greg KH <gregkh-AT-linuxfoundation.org>
To:	Konstantin Ryabitsev <konstantin-AT-linuxfoundation.org>
Subject:	[PATCH] 4.4 will be the next longterm kernel
Date:	Thu, 29 Oct 2015 05:41:05 +0900
Message-ID:	<20151028204105.GA13060@kroah.com>
Cc:	linux-kernel-AT-vger.kernel.org, stable-AT-vger.kernel.org
Archive-link:	Article

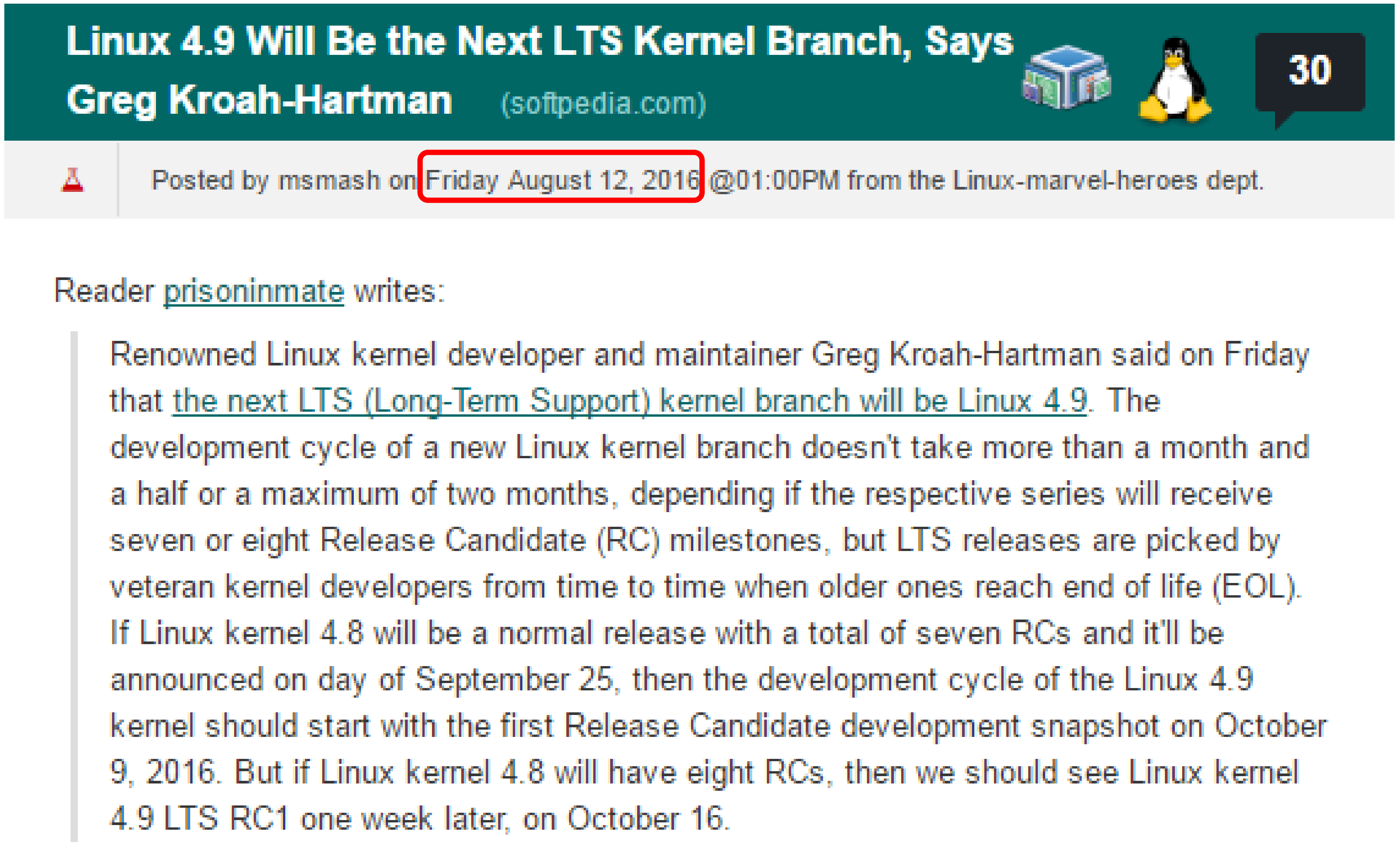
```
From: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

As a result of the discussion at the 2015 Linux Kernel summit, the 4.4
kernel will be the next longterm kernel release, so notify everyone
about this on the web site.

Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
---
 content/releases.rst | 1 +
 1 file changed, 1 insertion(+)
```

https://lwn.net/Articles/662973/c

There is no LTSI-4.4 (We skipped this year’s LTSI)

- Community LTS
  - 3.10 (2014)
  - 4.1（2015）
  - 4.4（2016） ---- we skipped 4.4-LTSI, CIP adopted 4.4 though!

- Release timing
  - LTS/LTSI candidate announce
    - 3.10, 3.14 announcement at LinuxCon JP timing (May/June timing)
    - 4.4 decision announced at kernel summit 2015 accidentally
  - LTSI merge window
  - LTSI release



Linux 4.9 Will Be the Next LTS Kernel Branch, Says Greg Kroah-Hartman (softpedia.com)

30

Posted by msmash on Friday August 12, 2016 @01:00PM from the Linux-marvel-heroes dept.

Reader prisoninmate writes:

> Renowned Linux kernel developer and maintainer Greg Kroah-Hartman said on Friday that the next LTS (Long-Term Support) kernel branch will be Linux 4.9. The development cycle of a new Linux kernel branch doesn't take more than a month and a half or a maximum of two months, depending if the respective series will receive seven or eight Release Candidate (RC) milestones, but LTS releases are picked by veteran kernel developers from time to time when older ones reach end of life (EOL). If Linux kernel 4.8 will be a normal release with a total of seven RCs and it'll be announced on day of September 25, then the development cycle of the Linux 4.9 kernel should start with the first Release Candidate development snapshot on October 9, 2016. But if Linux kernel 4.8 will have eight RCs, then we should see Linux kernel 4.9 LTS RC1 one week later, on October 16.

I am a Linaro stable kernel maintainer. Our stable kernel is base on LTS plus much of upstream features backporting on them. Here is the detailed info of LSK: https://wiki.linaro.org/LSK
https://git.linaro.org/?p=kernel/linux-linaro-stable.git

These kind of backporting features are requested by many LSK members which most are leading ARM product vendors. LSK target on the feature backporting collaboration, to reduce the duplicate work on that. Current LTSI: https://ltsi.linuxfoundation.org/what-is-ltsi, has
similar target for backporting collocation. but there are still couples problems.

1, LTSI is focus on board support more than feature backporting

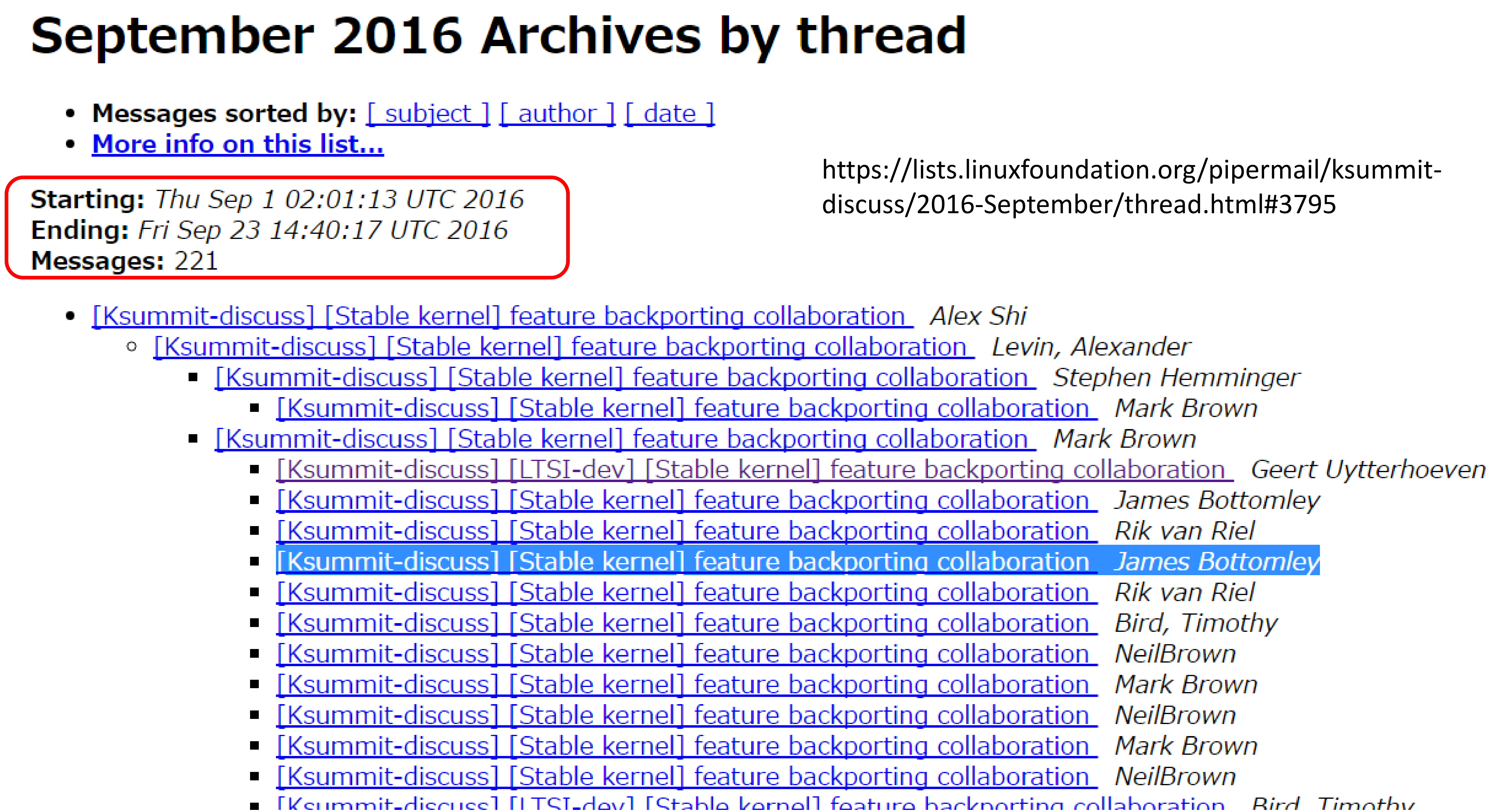
2, ltsi kernel version 3.10/3.14/4.1 is older than LTS and LSK 3.18/4.1/4.4.

3, merge everything together isn't good for some users and can not give user option to select preferred kernel feature. On the contrary, each of feature backported separately on latest LTS in LSK, user can just pick their wanted features and merge them for their own kernel.

4, all vendor specific driver in one branch get complains and developing status make it hard to handle changes in a fast-forward stable kernel.

As to LSK, although most feature are ARM related, but LSK also provide some common feature which works on other archs, like cgroupv2, RO-vDSO, KASAN, PAX_USERCOPY, etc. I believe this common backporting is also useful for common industries.
If so, could we call a better way for feature backporting collaboration?

Regards.
Alex



September 2016 Archives by thread

- **Messages sorted by:** [subject] [author] [date]
- **More info on this list...**

https://lists.linuxfoundation.org/pipermail/ksummit-discuss/2016-September/thread.html#3795

Starting: *Thu Sep 1 02:01:13 UTC 2016*
Ending: *Fri Sep 23 14:40:17 UTC 2016*
Messages: 221

- [Ksummit-discuss] [Stable kernel] feature backporting collaboration *Alex Shi*
  - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *Levin, Alexander*
    - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *Stephen Hemminger*
      - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *Mark Brown*
    - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *Mark Brown*
      - [Ksummit-discuss] [LTSI-dev] [Stable kernel] feature backporting collaboration *Geert Uytterhoeven*
      - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *James Bottomley*
      - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *Rik van Riel*
      - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *James Bottomley*
      - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *Rik van Riel*
      - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *Bird, Timothy*
      - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *NeilBrown*
      - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *Mark Brown*
      - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *NeilBrown*
      - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *Mark Brown*
      - [Ksummit-discuss] [Stable kernel] feature backporting collaboration *NeilBrown*
      - [Ksummit-discuss] [LTSI-dev] [Stable kernel] feature backporting collaboration *Bird, Timothy*

[Ksummit-discuss] [LTSI-dev] [Stable kernel] feature backporting collaboration

Theodore Ts'o tytso at mit.edu
Tue Sep 13 03:14:37 UTC 2016

So Greg has already said that if people abuse the preannouncement by trying to push obviously unready code into 4.9 to comply with enterprise distributions requirements that features have to be upstream first (although obviously the distributions would accept "bug fix" patches), he reserved the right to retroactively declare that 4.8 or 4.10 would be the LTS kernel.

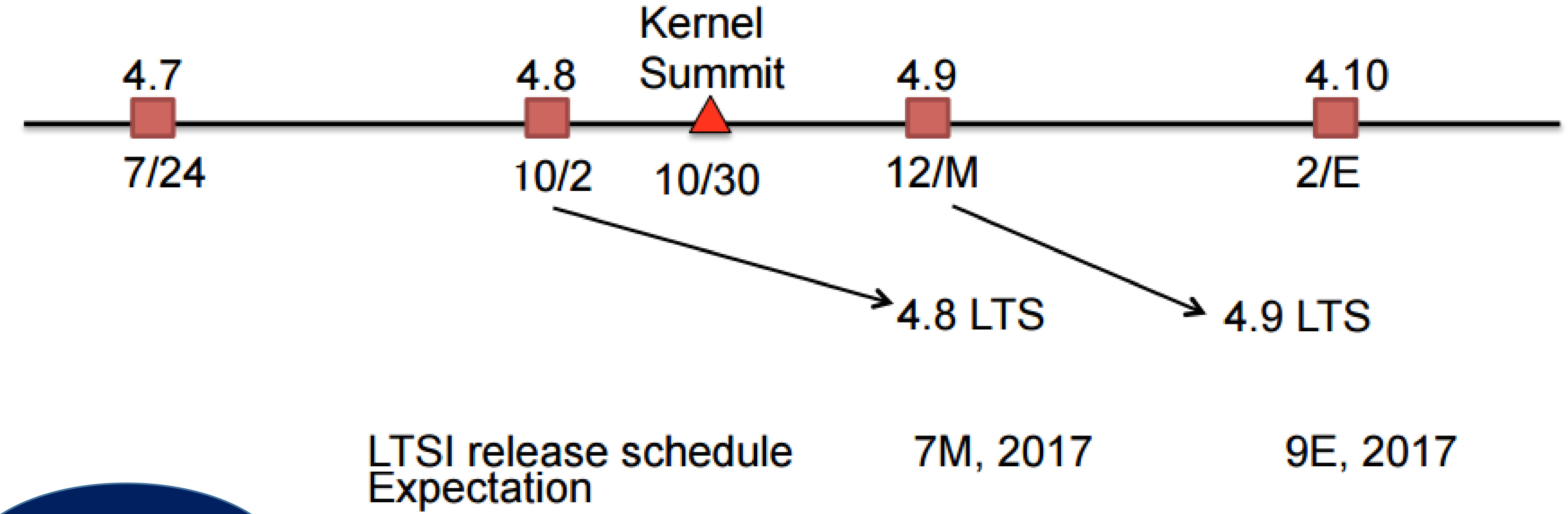
So even this year, if people behave badly (which is the reason why the announcement was done after the release -- people were trying to game the system) it is not guaranteed that 4.9 will be the LTS kernel.

- Ted

LTSI Schedule expectation

LONG TERM SUPPORT INITIATIVE

- Kernel Summit 2016 will be the time to decide LTS version



ELCE2016
Oct 2016

http://ltsi.linuxfoundation.org/sites/ltsi/files/LTSI-ELCE2016.pdf

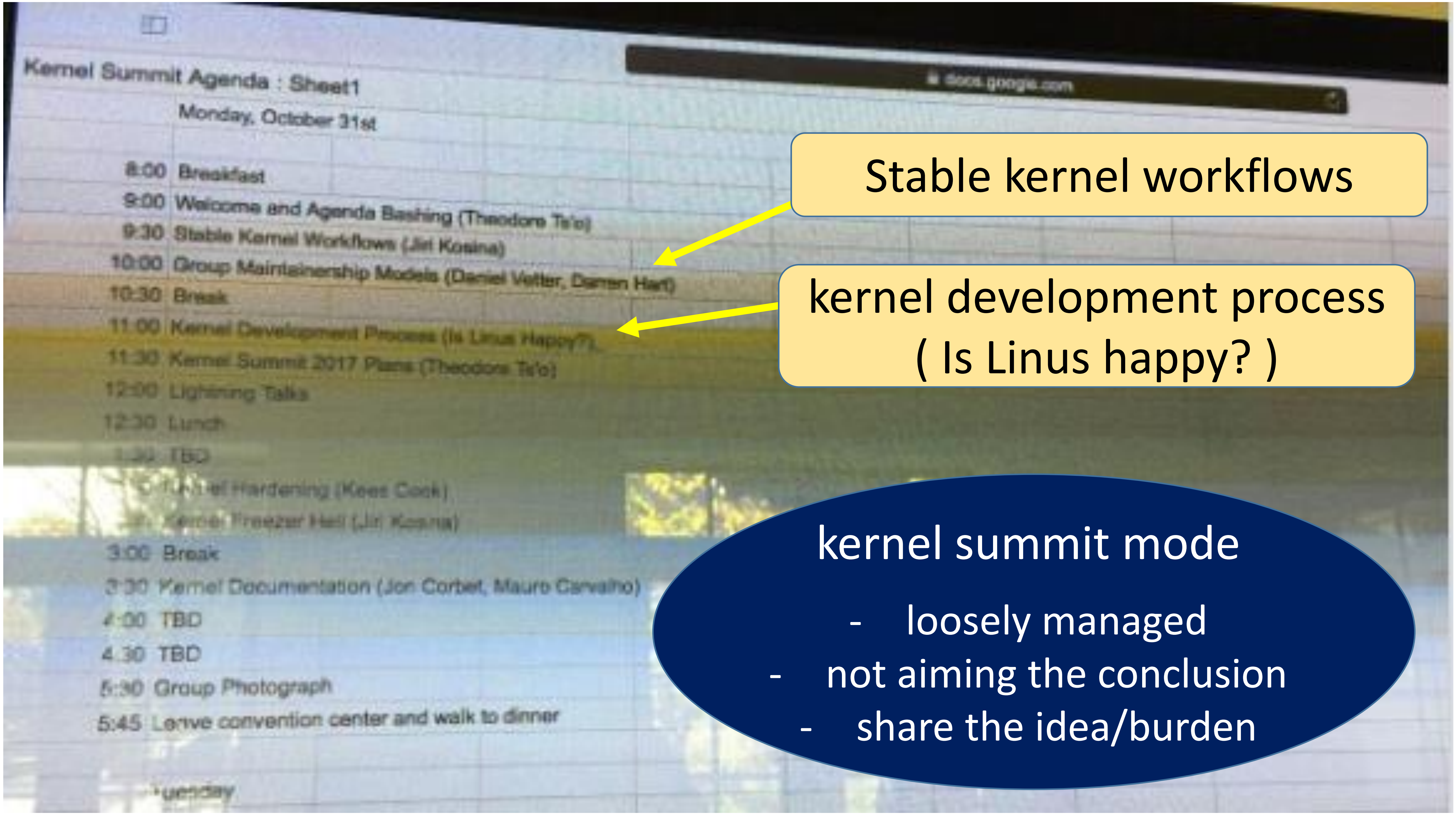
The 2016 Kernel Summit group photo

[Posted November 1, 2016 by corbet]

KS 2016
Nov 2016

THE ANNUAL LINUX
KERNEL SUMMIT
2016
Santa Fe, New Mexico
Core Invitation Day
THE LINUX FOUNDATION

THE ANNUAL LINUX
KERNEL SUMMIT
2016
Santa Fe, New Mexico
Thank You To Our Sponsors
General Sponsors
IBM
jumptrading
redhat
RENESAS
vmware
WD Western Digital
Breakfast & Breaks Sponsor


Kernel Summit Agenda : Sheet1

docs.google.com

Monday, October 31st

8:00	Breakfast
9:00	Welcome and Agenda Bashing (Theodore Ts'o)
9:30	Stable Kernel Workflows (Jiri Kosina)
10:00	Group Maintainership Models (Daniel Vetter, Darren Hart)
10:30	Break
11:00	Kernel Development Process (Is Linus Happy?)
11:30	Kernel Summit 2017 Plans (Theodore Ts'o)
12:00	Lightning Talks
12:30	Lunch
1:30	TBD
[illegible]	Kernel Hardening (Kees Cook)
[illegible]	Kernel Freezer Hell (Jiri Kosina)
3:00	Break
3:30	Kernel Documentation (Jon Corbet, Mauro Carvalho)
4:00	TBD
4:30	TBD
5:30	Group Photograph
5:45	Leave convention center and walk to dinner

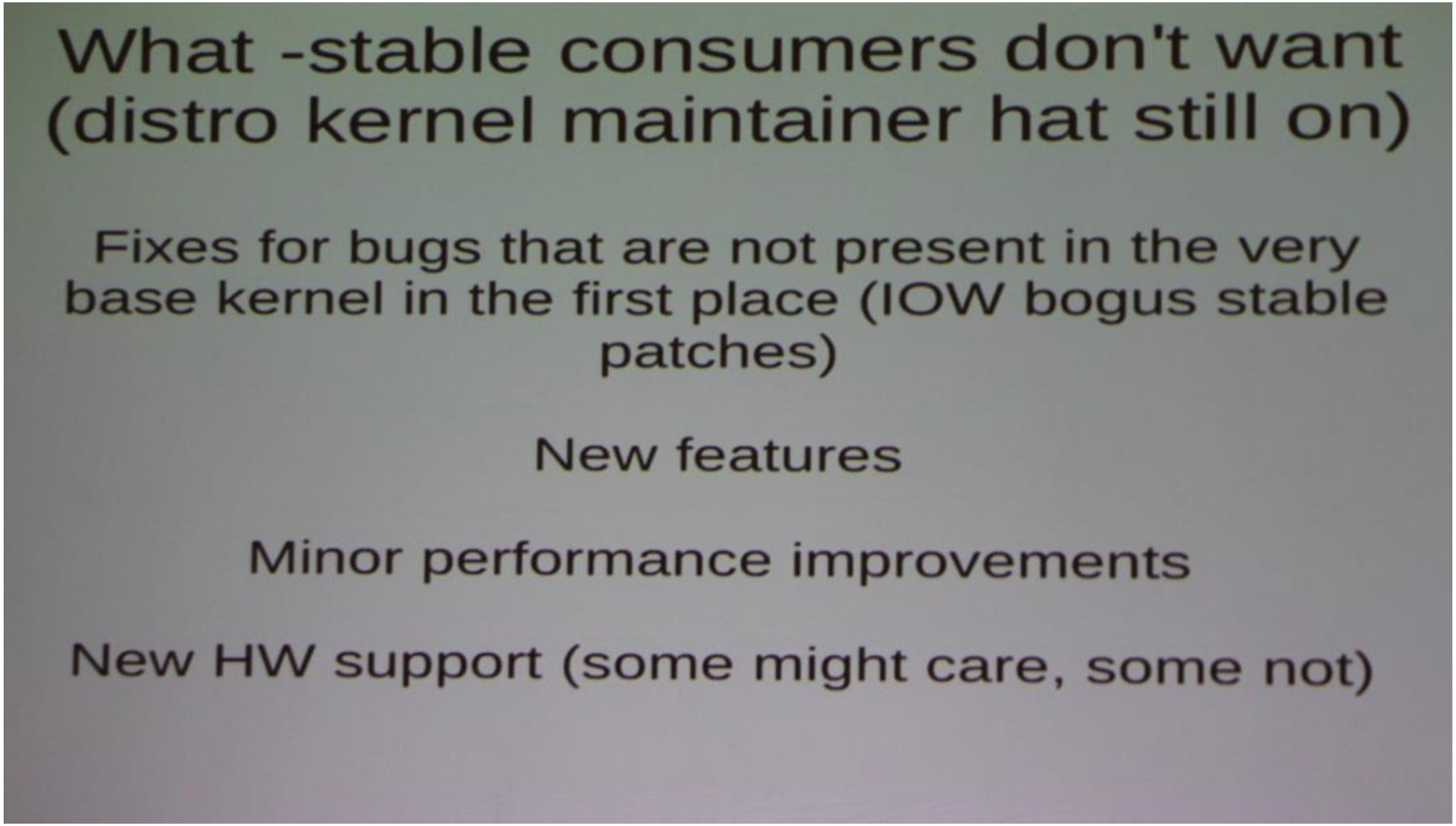
Tuesday

Stable kernel workflows

kernel development process
(Is Linus happy?)

kernel summit mode

- loosely managed
- not aiming the conclusion
- share the idea/burden



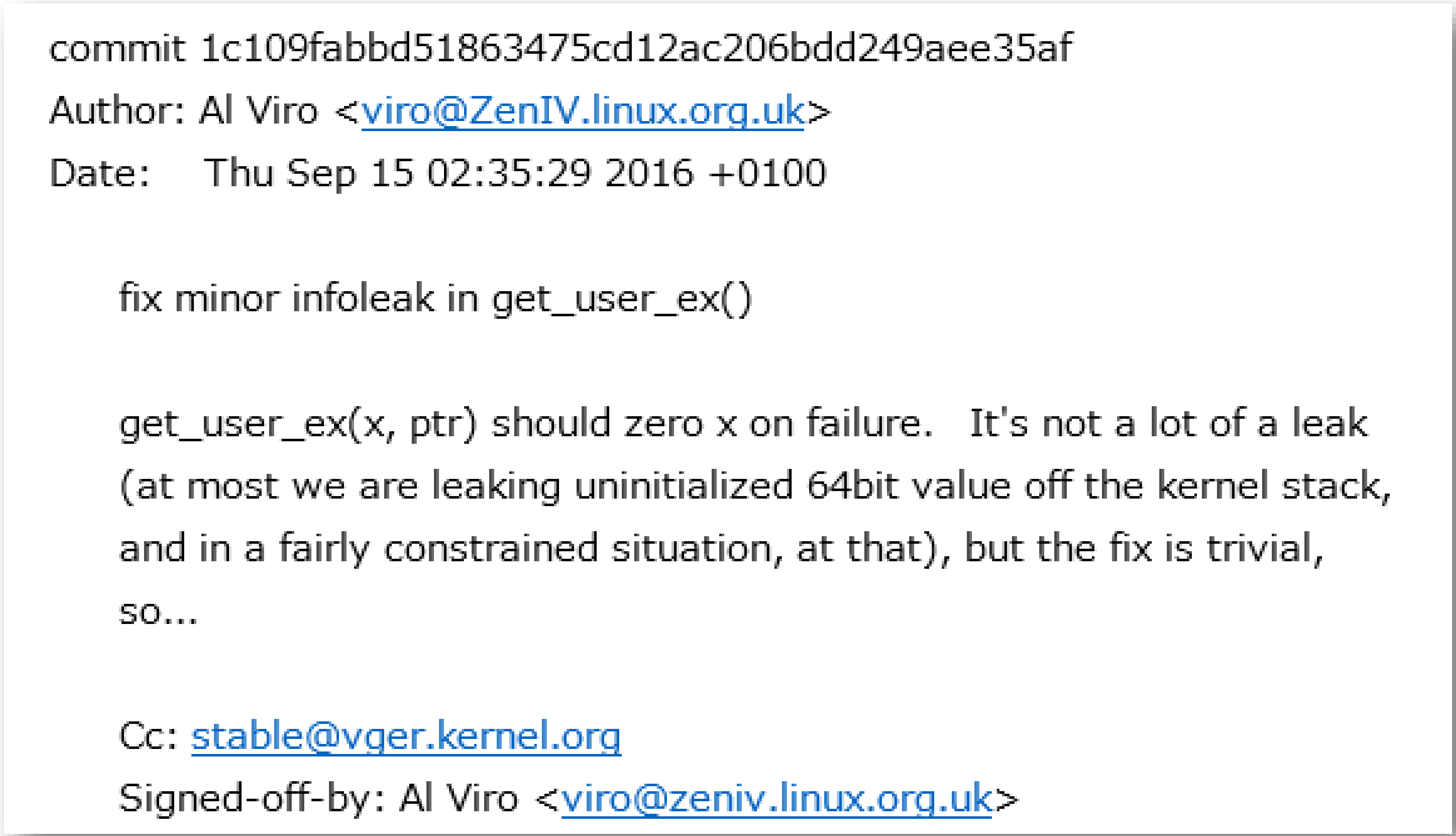
What -stable consumers don't want (distro kernel maintainer hat still on)

Fixes for bugs that are not present in the very base kernel in the first place (IOW bogus stable patches)

New features

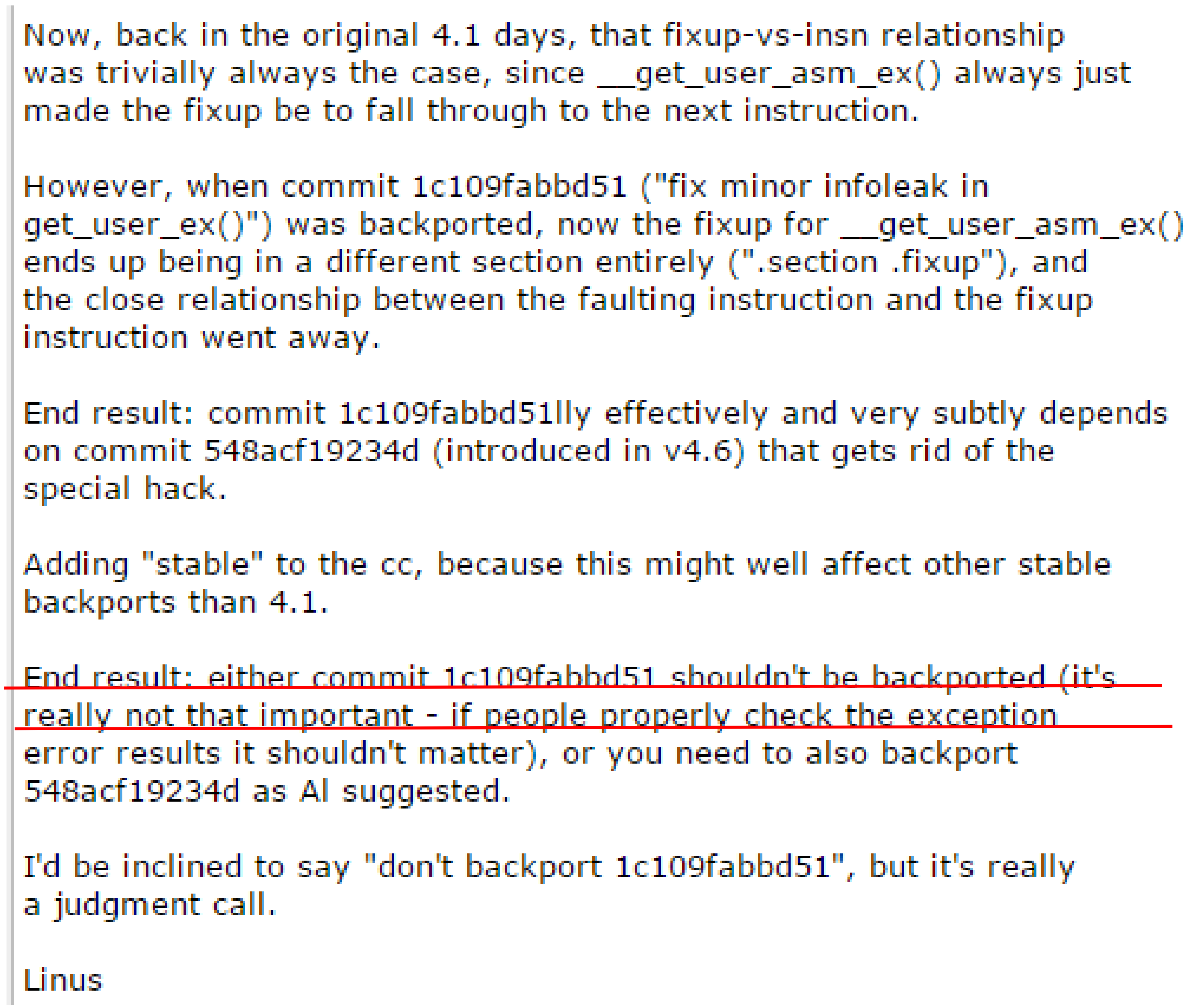
Minor performance improvements

New HW support (some might care, some not)



```
commit 1c109fabbd51863475cd12ac206bdd249aee35af
Author: Al Viro <viro@ZenIV.linux.org.uk>
Date:    Thu Sep 15 02:35:29 2016 +0100

    fix minor infoleak in get_user_ex()

    get_user_ex(x, ptr) should zero x on failure.   It's not a lot of a leak
    (at most we are leaking uninitialized 64bit value off the kernel stack,
    and in a fairly constrained situation, at that), but the fix is trivial,
    so...

    Cc: stable@vger.kernel.org
    Signed-off-by: Al Viro <viro@zeniv.linux.org.uk>
```

discussed in LKML at http://www.gossamer-threads.com/lists/linux/kernel/2555778.



Now, back in the original 4.1 days, that fixup-vs-insn relationship
was trivially always the case, since __get_user_asm_ex() always just
made the fixup be to fall through to the next instruction.

However, when commit 1c109fabbd51 ("fix minor infoleak in
get_user_ex()") was backported, now the fixup for __get_user_asm_ex()
ends up being in a different section entirely (".section .fixup"), and
the close relationship between the faulting instruction and the fixup
instruction went away.

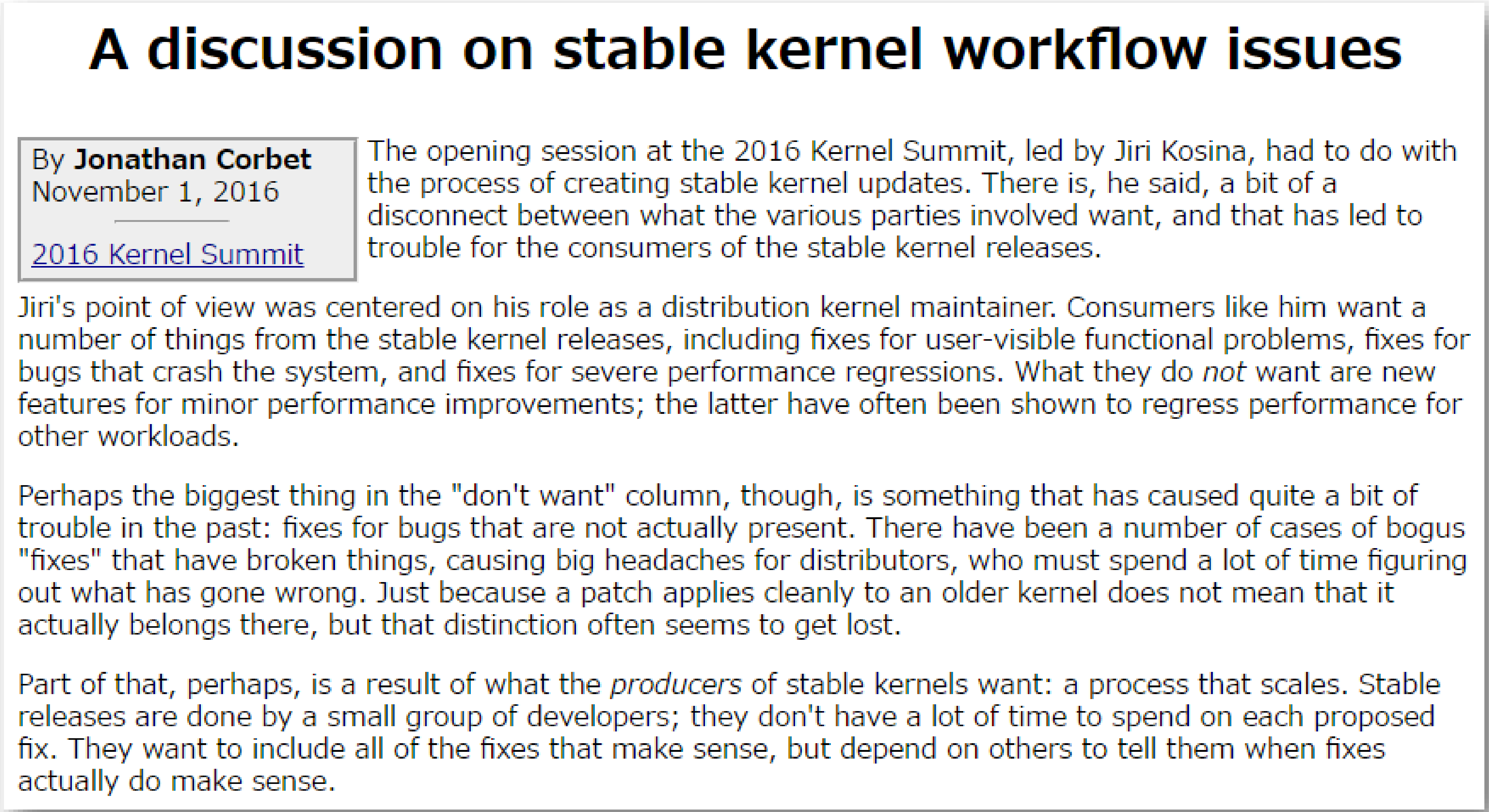
End result: commit 1c109fabbd51lly effectively and very subtly depends
on commit 548acf19234d (introduced in v4.6) that gets rid of the
special hack.

Adding "stable" to the cc, because this might well affect other stable
backports than 4.1.

End result: either commit 1c109fabbd51 shouldn't be backported (it's
really not that important - if people properly check the exception
error results it shouldn't matter), or you need to also backport
548acf19234d as Al suggested.

I'd be inclined to say "don't backport 1c109fabbd51", but it's really
a judgment call.

Linus



A discussion on stable kernel workflow issues

By **Jonathan Corbet**
November 1, 2016

2016 Kernel Summit

The opening session at the 2016 Kernel Summit, led by Jiri Kosina, had to do with the process of creating stable kernel updates. There is, he said, a bit of a disconnect between what the various parties involved want, and that has led to trouble for the consumers of the stable kernel releases.

Jiri's point of view was centered on his role as a distribution kernel maintainer. Consumers like him want a number of things from the stable kernel releases, including fixes for user-visible functional problems, fixes for bugs that crash the system, and fixes for severe performance regressions. What they do *not* want are new features for minor performance improvements; the latter have often been shown to regress performance for other workloads.

Perhaps the biggest thing in the "don't want" column, though, is something that has caused quite a bit of trouble in the past: fixes for bugs that are not actually present. There have been a number of cases of bogus "fixes" that have broken things, causing big headaches for distributors, who must spend a lot of time figuring out what has gone wrong. Just because a patch applies cleanly to an older kernel does not mean that it actually belongs there, but that distinction often seems to get lost.

Part of that, perhaps, is a result of what the *producers* of stable kernels want: a process that scales. Stable releases are done by a small group of developers; they don't have a lot of time to spend on each proposed fix. They want to include all of the fixes that make sense, but depend on others to tell them when fixes actually do make sense.

https://lwn.net/Articles/705220/