Linux Kernel Release Model

(and security stuff)

Greg Kroah-Hartman
gregkh@linuxfoundation.org

github.com/gregkh/presentation-release-model

60,000 files
24,767,000 lines

Kernel release 4.13.0

4,316 developers
519 companies

Kernel releases 4.8.0 – 4.13.0
August 2016 – September 2017

10,000 lines added

2,500 lines removed

2,100 lines modified

Kernel releases 4.8.0 – 4.13.0
August 2016 – September 2017

10,000 lines added
2,500 lines removed
2,100 lines modified

Every day

Kernel releases 4.8.0 – 4.13.0
August 2016 – September 2017

8.5 changes per hour

Kernel releases 4.8.0 – 4.13.0
August 2016 – September 2017

9.7 changes per hour

4.9 & 4.12 release

Old release model

2.2 – January 1999
2.4 – January 2001
2.6 – December 2003

“New” release model

Release every 2-3 months
All releases are stable

“Cambridge Promise”

Will not break userspace.

“Cambridge Promise”

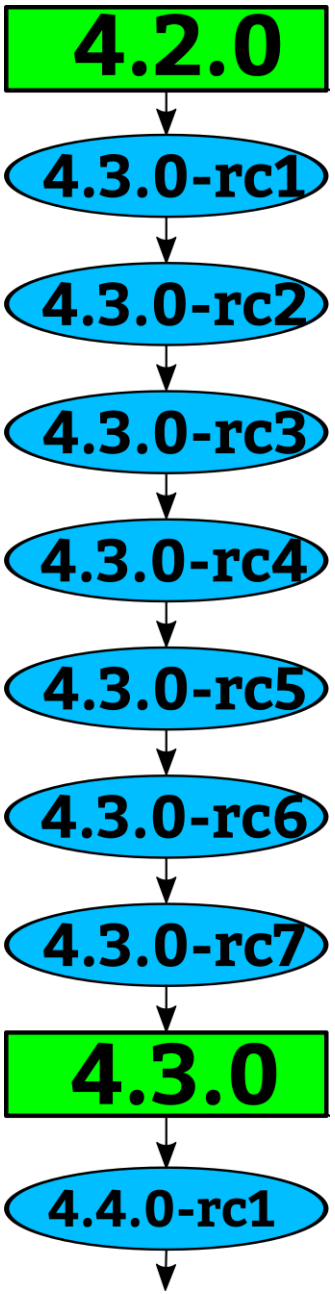
Will not break userspace, knowingly.

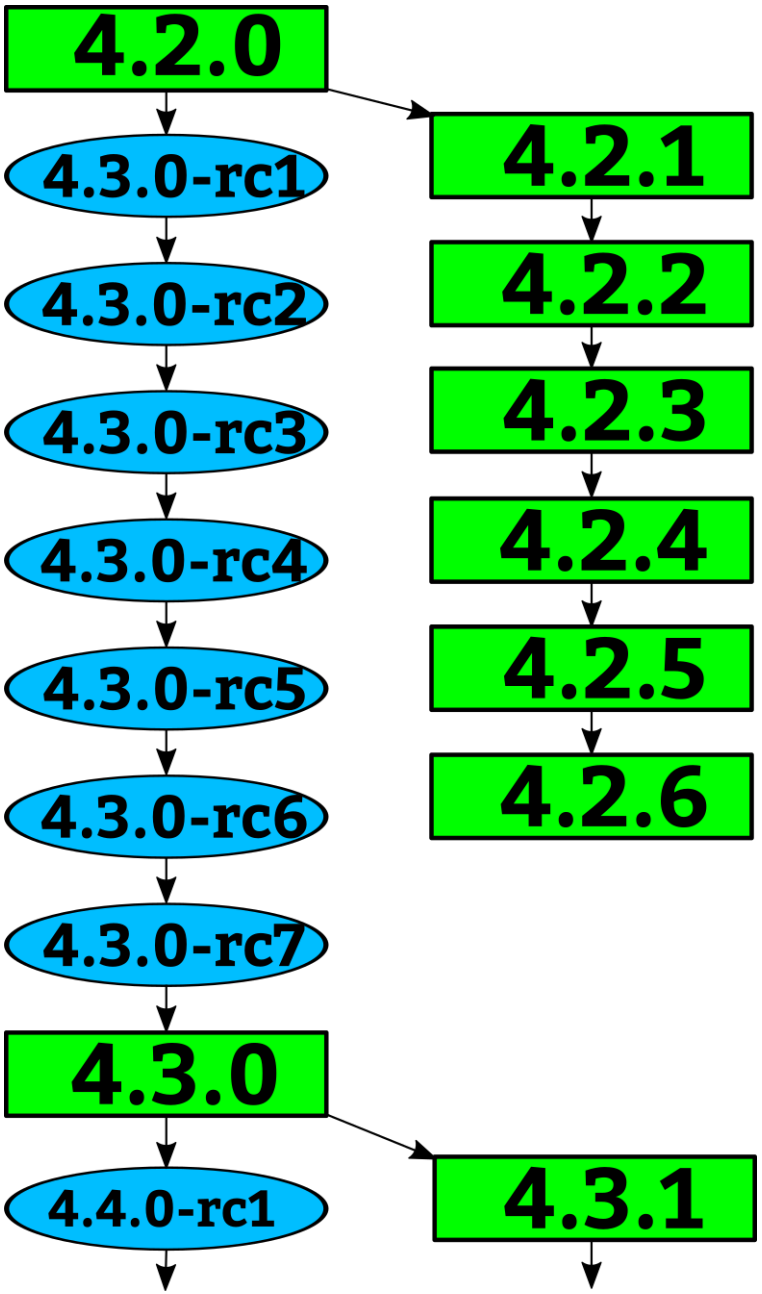
– July 2007

Version numbers mean nothing

2.6.x → 3.x 2011

3.x → 4.x 2015


4.2.0
4.3.0-rc1
4.3.0-rc2
4.3.0-rc3
4.3.0-rc4
4.3.0-rc5
4.3.0-rc6
4.3.0-rc7
4.3.0
4.4.0-rc1


4.2.0
4.3.0-rc1
4.3.0-rc2
4.3.0-rc3
4.3.0-rc4
4.3.0-rc5
4.3.0-rc6
4.3.0-rc7
4.3.0
4.4.0-rc1
4.2.1
4.2.2
4.2.3
4.2.4
4.2.5
4.2.6
4.3.1

Stable rules

- Bugfix
- Less than 100 lines
- New ids or quirks
- Must be in Linus's tree

https://www.kernel.org/doc/html/latest/process/stable-kernel-rules.html

“Longterm kernels”

One picked per year
Maintained for at least 2 years

4.4 4.9 4.14

4.4	9 changes / day
4.9	13 changes / day
4.12	10 changes / day

Every release is stable

Decade old guarantee

Always update your kernel

Can’t update your kernel?

Blame your SoC provider...

“Popular” SoC kernel tree

```
6171 files changed 2837180 insertions(+), 42568 deletions(-)
```

“Popular” SoC kernel tree

```
6171 files changed 2837180 insertions(+), 42568 deletions(-)
```

`Image.lz4` – 3.2 million lines

Linux “like”

Kernel Security

Kernel Security

Almost all bugs can be a “security” issue.

Kernel Security

Almost all bugs can be a “security” issue.

Fix them as soon as possible.

Linus wrote in 2008:

On Wed, 16 Jul 2008, pageexec@freemail.hu wrote:
>
> you should check out the last few -stable releases then and see how
> the announcement doesn't ever mention the word 'security' while fixing
> security bugs

Umm. What part of "they are just normal bugs" did you have issues with?
I expressly told you that security bugs should not be marked as such,
because bugs are bugs.

> in other words, it's all the more reason to have the commit say it's
> fixing a security issue.

No.

> > I'm just saying that why mark things, when the marking have no meaning?
> > People who believe in them are just _wrong_.
>
> what is wrong in particular?

You have two cases:

- people think the marking is somehow trustworthy.

  People are WRONG, and are misled by the partial markings, thinking that unmarked bugfixes are "less important". They aren't.

- People don't think it matters

  People are right, and the marking is pointless.
  In either case it's just stupid to mark them. I don't want to do it, because I don't want to perpetuate the myth of "security fixes" as a separate thing from "plain regular bug fixes".

  They're all fixes. They're all important. As are new features, for that matter.

> when you know that you're about to commit a patch that fixes a security
> bug, why is it wrong to say so in the commit?

It's pointless and wrong because it makes people think that other bugs
aren't potential security fixes.

What was unclear about that?

Linus

Above email:

http://marc.info/?l=linux-kernel&m=121616463003140&w=2

Whole thread:

http://marc.info/?t=121507404600023&r=4&w=2

Reported security issue:

On Wed, 16 Jul 2008, pageexec@freemail.hu wrote:
>
> we went through this and you yourself said that security bugs are *not*
> treated as normal bugs because you do omit relevant information from such
> commits

Actually, we disagree on one fundamental thing. We disagree on
that single word: "relevant".

I do not think it's helpful _or_ relevant to explicitly point out how to
tigger a bug. It's very helpful and relevant when we're trying to chase
the bug down, but once it is fixed, it becomes irrelevant.

You think that explicitly pointing something out as a security issue is
really important, so you think it's always "relevant". And I take mostly
the opposite view. I think pointing it out is actually likely to be
counter-productive.

For example, the way I prefer to work is to have people send me and the kernel list a patch for a fix, and then in the very next email send (in private) an example exploit of the problem to the security mailing list (and that one goes to the private security list just because we don't want all the people at universities rushing in to test it). THAT is how things should work.

Should I document the exploit in the commit message? Hell no. It's private for a reason, even if it's real information. It was real information for the developers to explain why a patch is needed, but once explained, it shouldn't be spread around unnecessarily.

Linus

Above email:

http://marc.info/?l=linux-kernel&m=121616807207387&w=2

Reporting Security bugs:

https://www.kernel.org/doc/html/latest/admin-guide/security-bugs.html

Keeping a Secure System

Take **all** stable kernel updates

Enable hardening features

“If you are not using a stable / longterm kernel, your machine is insecure”

– me

“Ceaseless change is the only constant thing in Nature.”

– John Candee Dean

github.com/gregkh/presentation-release-model

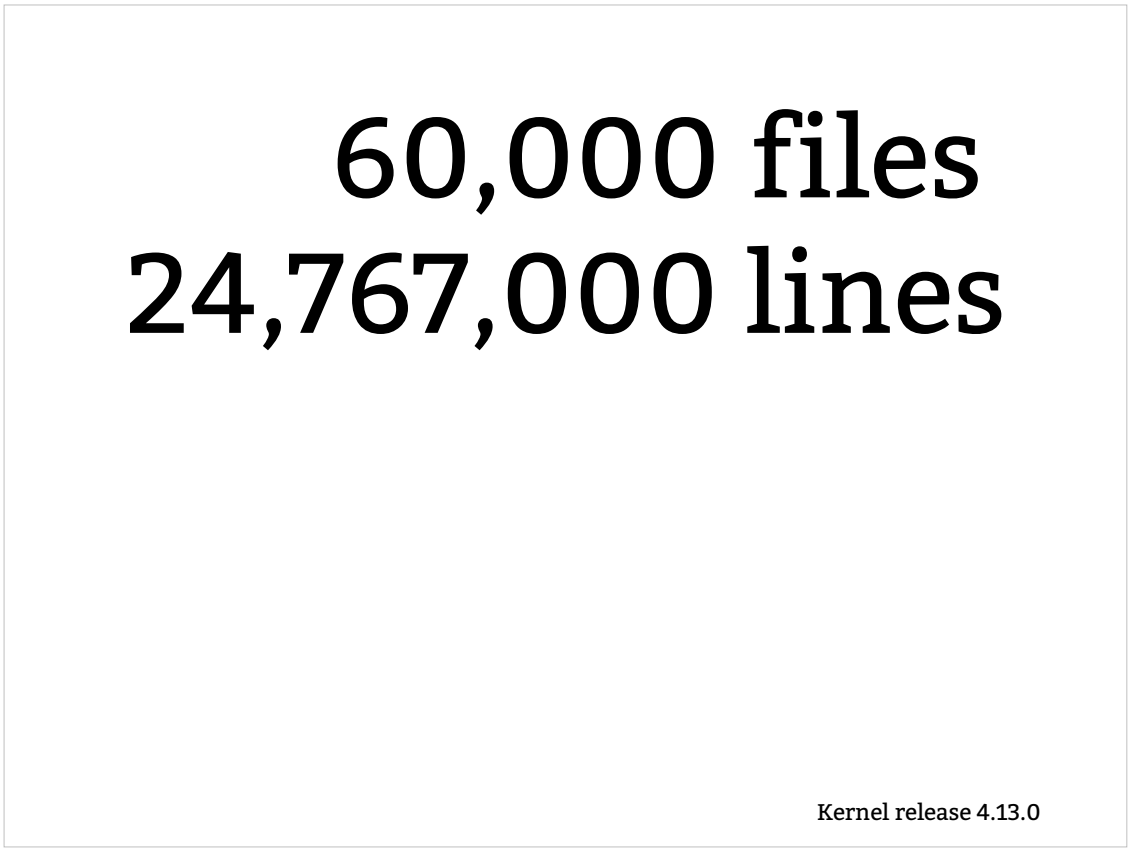
Linux Kernel Release Model

(and security stuff)

Greg Kroah-Hartman
gregkh@linuxfoundation.org

github.com/gregkh/presentation-release-model

I'm going to discuss the how fast the kernel is moving, how we do it all, and how you can get involved.



This was for the 4.11 kernel release, which happened April 30, 2017.

4,316 developers
519 companies

Kernel releases 4.8.0 – 4.13.0
August 2016 – September 2017

This makes the Linux kernel the largest contributed body of software out there that we know of.

This is just the number of companies that we know about, there are more that we do not, and as the responses to our inquiries come in, this number will go up.

Have surpassed 400 companies for 4 years now.

10,000 lines added
2,500 lines removed
2,100 lines modified

Kernel releases 4.8.0 – 4.13.0
August 2016 – September 2017

10,000 lines added
2,500 lines removed
2,100 lines modified

Every day

Kernel releases 4.8.0 – 4.13.0
August 2016 – September 2017

8.5 changes per hour

Kernel releases 4.8.0 – 4.13.0
August 2016 – September 2017

This is 24 hours a day, 7 days a week, for a full year.

We went this fast the year before this as well, this is an amazing rate of change.

Interesting note, all of these changes are all through the whole kernel.

For example, the core kernel is only 5% of the code, and 5% of the change was to the core kernel. Drivers are 55%, and 55% was done to them, it's completely proportional all across the whole kernel.

9.7 changes per hour

4.9 & 4.12 release

4.9 was the “largest” in number of changes that we have ever accepted. After 4.9, things went down a bit for 4.10 and 4.11, but 4.12 is getting very big.

Now this is just the patches we accepted, not all of the patches that have been submitted, lots of patches are rejected, as anyone who has ever tried to submit a patch can attest to.

Old release model

2.2 – January 1999
2.4 – January 2001
2.6 – December 2003

“New” release model

Release every 2-3 months
All releases are stable

“Cambridge Promise”

Will not break userspace.

Kernel summit 2007 in Cambridge England

All kernel developers agreed that this is what we will do, in order to give users a reason to feel comfortable upgrading their kernels.

“Cambridge Promise”

Will not break userspace,
knowingly.

– July 2007

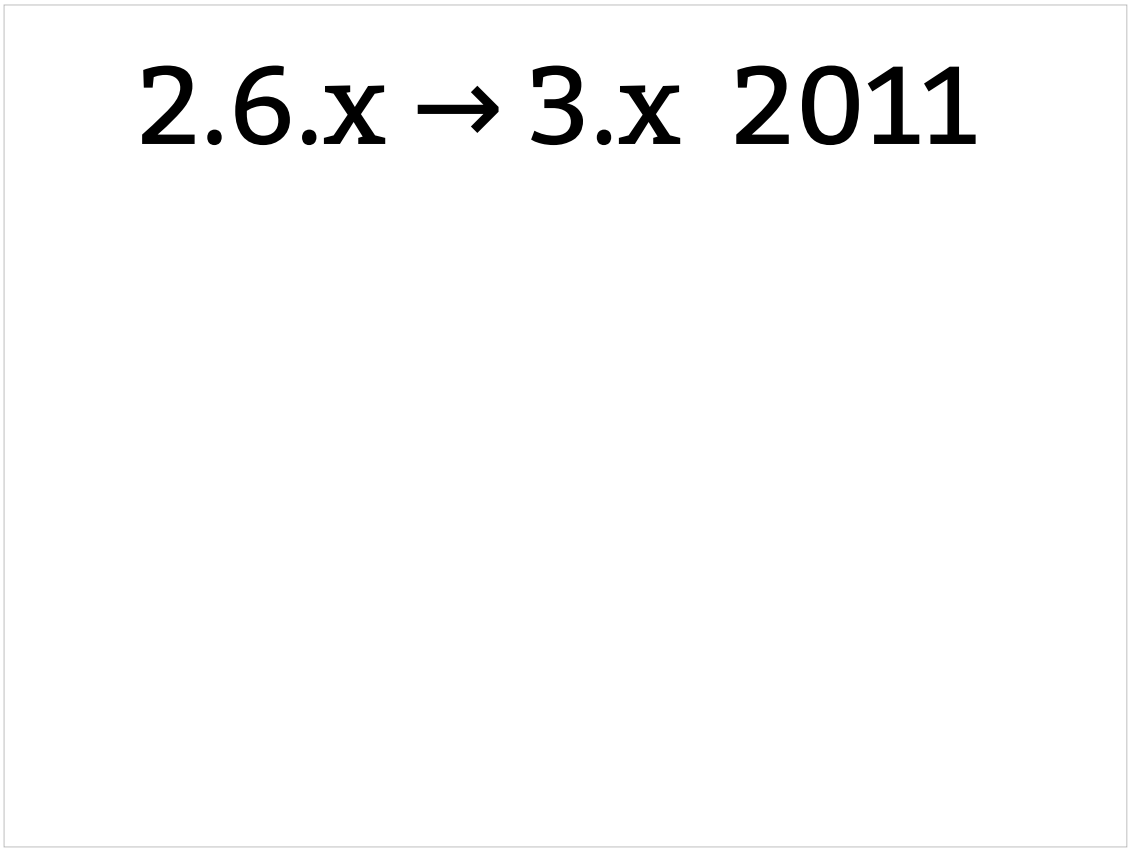
Well, we do not knowingly break userspace, we accidentally do it all the time, we are just human.

But we will work very hard to fix the issue.

Note, if no one notices userspace is broken, it isn’t.

Version numbers mean nothing

They only mean that one is newer than another.

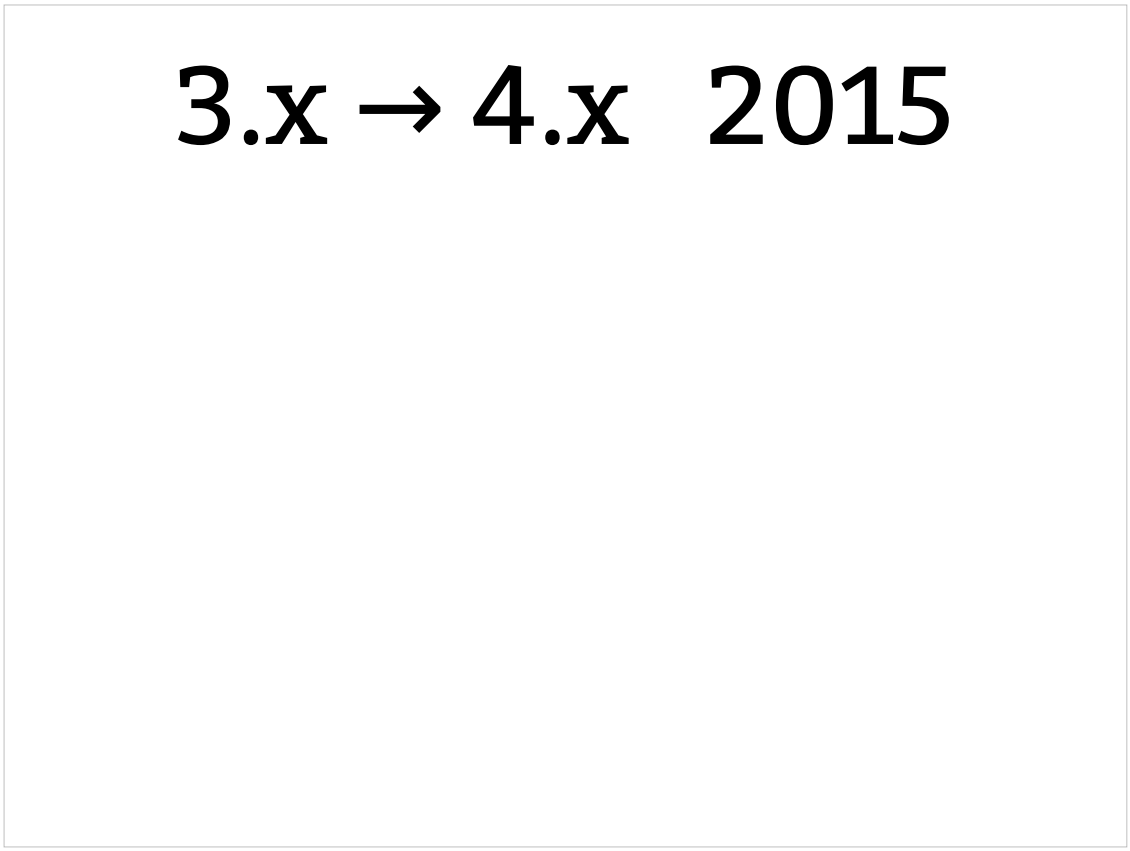


Big numbers seem to increment “smaller” over time than small numbers (brains are wierd)

LinuxCon Japan 2011

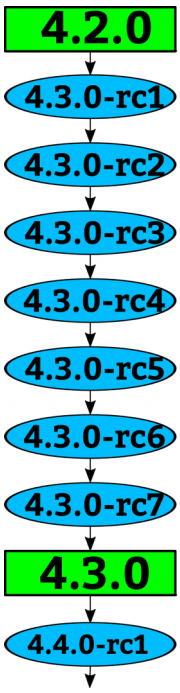
I bribed Linus with whisky

Kernel developers drank the bottle within minutes at the after-party.

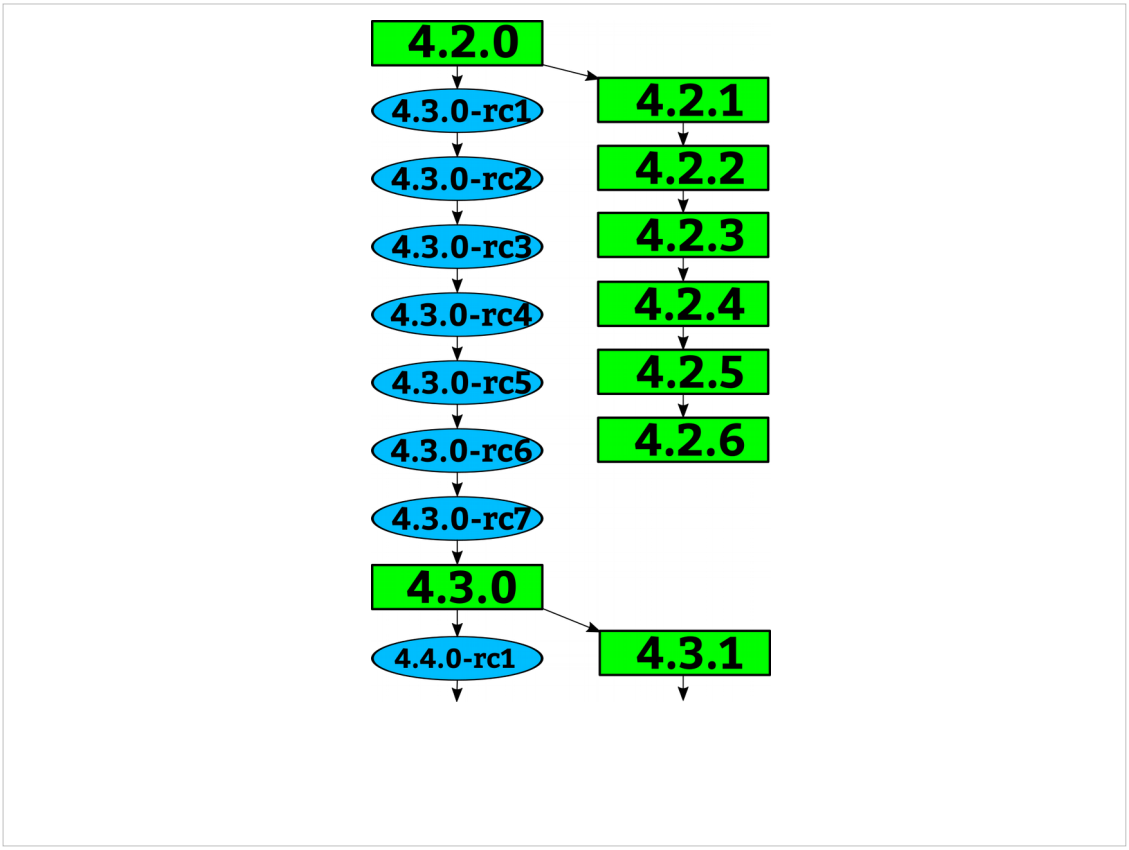


Big numbers seem to increment “smaller” over time than small numbers (brains are wierd)

2015



How a kernel is developed.
Linus releases a stable kernel
- 2 week merge window from subsystem maintainers
- rc1 is released
- bugfixes only now
- 2 weeks later, rc2
- bugfixes and regressions
- 2 weeks later,rc3
And so on until all major bugfixes and regressions are resolved and then the cycle starts over again.



Greg takes the stable releases from Linus, and does stable releases with them, applying only fixes that are already in Linus's tree.

Requiring fixes to be in Linus's tree first ensures that there is no divergence in the development model.

After Linus releases a new stable release, the old stable series is dropped.

With the exception of “longterm” stable releases, those are special, the stick around for much longer...

Stable rules

- Bugfix
- Less than 100 lines
- New ids or quirks
- Must be in Linus’s tree

https://www.kernel.org/doc/html/latest/process/stable-kernel-rules.html

They only mean that one is newer than another.

“Longterm kernels”

One picked per year
Maintained for at least 2 years

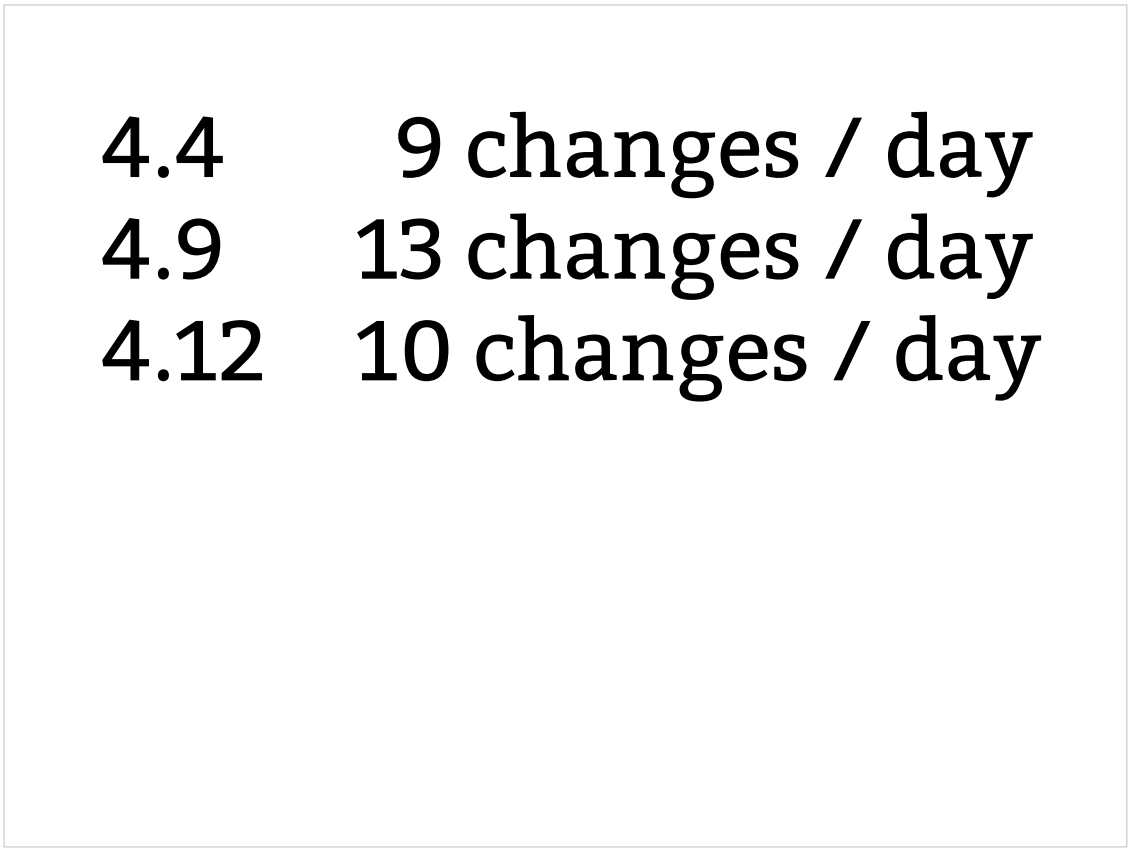
4.4 4.9 4.14

I pick one kernel release per year to maintain for longer than one release cycle. This kernel I will maintain for at least 2 years.

This means there are 2 longterm kernels being maintained at the same time.

4.4 and 4.9 are the longterm kernel releases I am currently maintaining

The LTSI project is based on the longterm kernels.



Lots of changes are getting backported

3.10 – averaging 4.5 a day
3.18 – 5 / day

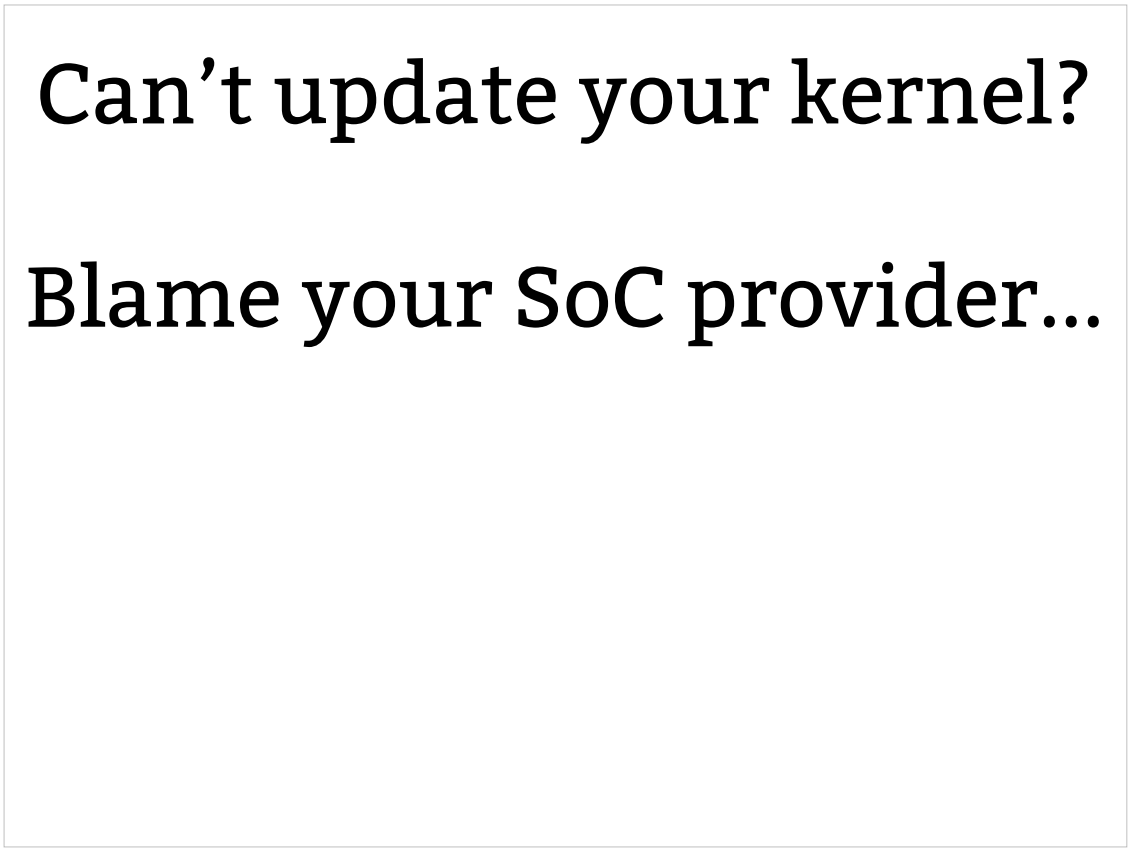
3.16 – 2 / day (debian is tough work...)

Every release is stable

Decade old guarantee

Always update your kernel

Wait what? Why update?



SoC kernels suck ass.

“Popular” SoC kernel tree

```
6171 files changed 2837180 insertions(+), 42568 deletions(-)
```

SoC kernels suck ass.

“Popular” SoC kernel tree

```
6171 files changed 2837180 insertions(+), 42568 deletions(-)
```

`Image.lz4` – 3.2 million lines

Linux “like”

SoC kernels suck ass.

88% of your kernel has never been reviewed by anyone in the community…

Kernel Security

Let’s talk about kernel security.

Kernel Security

Almost all bugs can be a “security” issue.

Anything that goes wrong in the kernel can usually be turned into a “security” problem.

Be it a DoS, or a reboot, or local root exploit, or worst case, a remote root exploit (very rare, thankfully.)

Kernel Security

Almost all bugs can be a “security” issue.

Fix them as soon as possible.

Because it’s really hard to determine if a bug is a “security” issue, our response is that we fix all bugs as soon as possible once we learn about them.

TTY bug in RH

Linus wrote in 2008:

```
On Wed, 16 Jul 2008, pageexec@freemail.hu wrote:
>
> you should check out the last few -stable releases then and see how
> the announcement doesn't ever mention the word 'security' while fixing
> security bugs

Umm. What part of "they are just normal bugs" did you have issues with?
I expressly told you that security bugs should not be marked as such,
because bugs are bugs.

> in other words, it's all the more reason to have the commit say it's
> fixing a security issue.

No.
```

Why we do this, Linus wrote:

> > I'm just saying that why mark things, when the marking have no meaning?
> > People who believe in them are just _wrong_.
>
> what is wrong in particular?

You have two cases:

- people think the marking is somehow trustworthy.

  People are WRONG, and are misled by the partial markings, thinking that
  unmarked bugfixes are "less important". They aren't.

- People don't think it matters

  People are right, and the marking is pointless.
  In either case it's just stupid to mark them. I don't want to do it,
  because I don't want to perpetuate the myth of "security fixes" as a
  separate thing from "plain regular bug fixes".

  They're all fixes. They're all important. As are new features, for that
  matter.

```
> when you know that you're about to commit a patch that fixes a security
> bug, why is it wrong to say so in the commit?

It's pointless and wrong because it makes people think that other bugs
aren't potential security fixes.

What was unclear about that?

Linus
```

Above email:
    http://marc.info/?l=linux-kernel&m=121616463003140&w=2

Whole thread:
    http://marc.info/?t=121507404600023&r=4&w=2

Go read the whole email thread (100 emails), it's a good summary of why we do what we do.

You also see me get mad at a user, rare...

Reported security issue:

On Wed, 16 Jul 2008, pageexec@freemail.hu wrote:
>
> we went through this and you yourself said that security bugs are *not*
> treated as normal bugs because you do omit relevant information from such
> commits

Actually, we disagree on one fundamental thing. We disagree on
that single word: "relevant".

I do not think it's helpful _or_ relevant to explicitly point out how to
tigger a bug. It's very helpful and relevant when we're trying to chase
the bug down, but once it is fixed, it becomes irrelevant.

You think that explicitly pointing something out as a security issue is
really important, so you think it's always "relevant". And I take mostly
the opposite view. I think pointing it out is actually likely to be
counter-productive.

For example, the way I prefer to work is to have people send me and the kernel list a patch for a fix, and then in the very next email send (in private) an example exploit of the problem to the security mailing list (and that one goes to the private security list just because we don't want all the people at universities rushing in to test it). THAT is how things should work.

Should I document the exploit in the commit message? Hell no. It's private for a reason, even if it's real information. It was real information for the developers to explain why a patch is needed, but once explained, it shouldn't be spread around unnecessarily.

Linus

Above email:

http://marc.info/?l=linux-kernel&m=121616807207387&w=2

Reporting Security bugs:

https://www.kernel.org/doc/html/latest/admin-guide/security-bugs.html

Keeping a Secure System

Take **all** stable kernel updates

Enable hardening features

Do not pick and choose!

The releases have been reviewed by the kernel developers as a whole, not in individual parts

It is hard, if not impossible, to determine which patches fix “security” issues and which do not. Almost every LTS release contains at least one known security fix, and many yet “unknown”.

If testing shows a problem, the kernel developer community will react quickly to resolve the issue. If you wait months or years to do an update, the kernel developer community will not be able to even remember what the updates were given the long delay.

Changes to parts of the kernel that you do not build/run are fine and can cause no problems to your system. To try to filter out only the changes you run will cause a kernel tree that will be impossible to merge correctly with future upstream

> “If you are not using a stable / longterm kernel, your machine is insecure”
>
> – me

Your infrastructure HAS to support updating the kernel. If you can’t do that, you are insecure.

Even the “enterprise” kernels aren’t keeping up with this rate of change, the exception being Debian.

If you use these kernels, you HAVE to keep up to date.

Android example demo!

“Ceaseless change is the only constant thing in Nature.”

– John Candee Dean

1911 astronomer.

If your operating system isn’t constantly changing, then it is dead. The world doesn’t stop changing, learn to embrace the change in order to survive.

“static systems” die.

Obligatory Penguin Picture