Linux Kernel Development

Going faster than you think

Greg Kroah-Hartman
gregkh@linuxfoundation.org
github.com/gregkh/kernel-development

53.000 files

21.150.000 lines

Kernel release 4.5.0

3.974 developers
440 companies

Kernel releases 4.0.0 – 4.5.0
April 2015 – March 2016

10.800 lines added

5.300 lines removed

1.875 lines modified

Kernel releases 4.0.0 – 4.5.0
April 2015 – March 2016

10.800 lines added

5.300 lines removed

1.875 lines modified

Every day

Kernel releases 4.0.0 – 4.5.0
April 2015 – March 2016

7,8 changes per hour

Kernel releases 4.0.0 – 4.5.0
April 2015 – March 2016

8,1 changes per hour

4.3 release

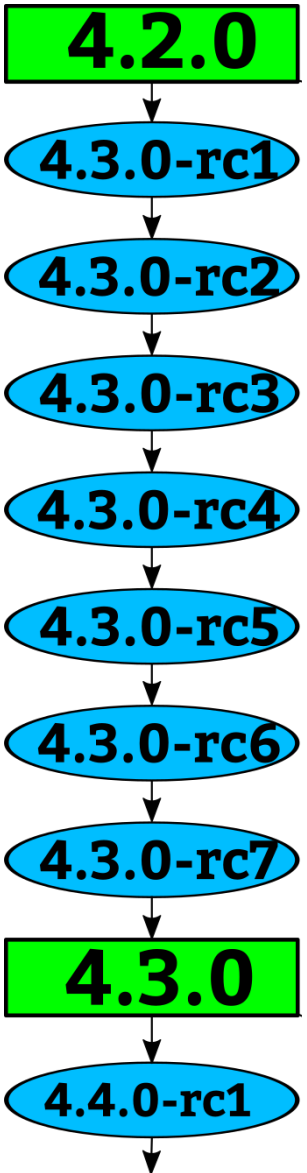
How we stay sane

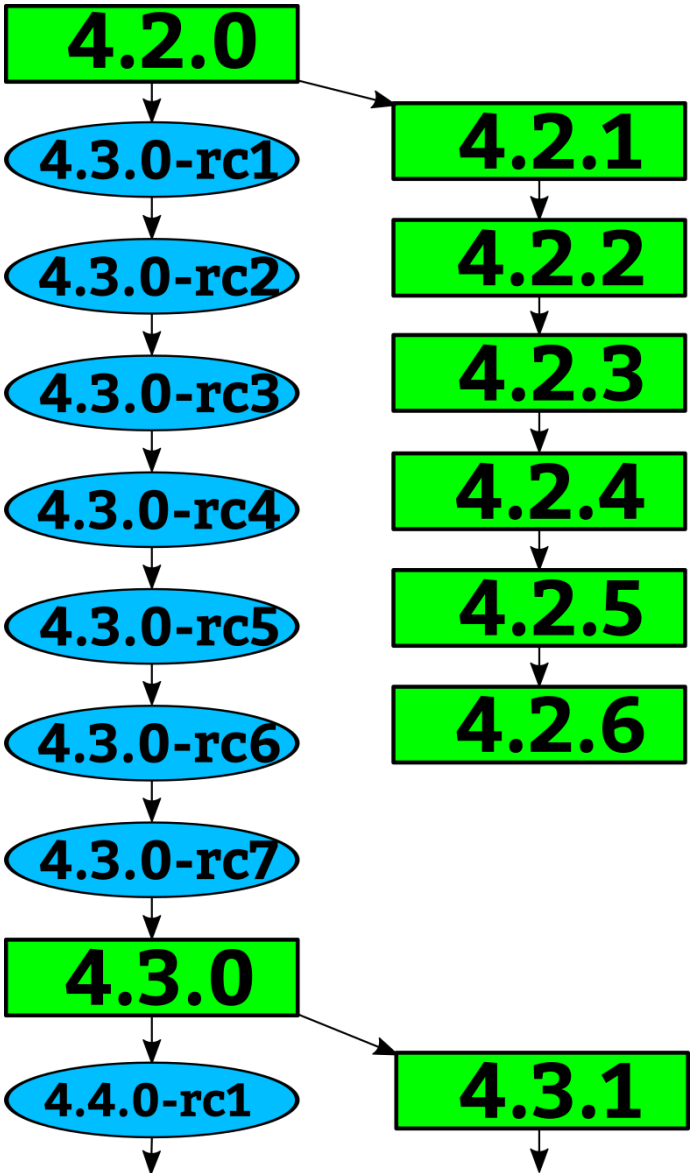
Time based releases

Incremental changes

New release every 2½ months

Kernel releases 3.10.0 – 4.5.0


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

“Longterm kernels”

One picked per year
Maintained for two years

3.14 4.1 4.4

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.

Kernel Security

Averaging 10 fixes per day.

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

– me

“The kernel needs airbags”
– Konstantin Ryabitsev

slides.com/mricon/giant-bags-of-mostly-water#/

“We will always have bugs,
we must stop their exploitation”
– Kees Cook

outflux.net/slides/2015/ks/security.pdf

Kernel Hardening

kernsec.org/wiki/index.php/Kernel_Self_Protection_Project

Core Infrastructure Initiative

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

– John Candee Dean

github.com/gregkh/kernel-development

Linux Kernel Development

Going faster than you think

Greg Kroah-Hartman
gregkh@linuxfoundation.org
github.com/gregkh/kernel-development

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

53.000 files
21.150.000 lines

Kernel release 4.5.0

This was for the 3.19 kernel release, which happened February 8, 2015.

The 3.17 kernel is the only release we have had in the past 4 years that we went down in size, this has only happened twice in the past 10 years.

3.974 developers
440 companies

Kernel releases 4.0.0 – 4.5.0
April 2015 – March 2016

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 2 years now.

10.800 lines added
5.300 lines removed
1.875 lines modified

Kernel releases 4.0.0 – 4.5.0
April 2015 – March 2016

10.800 lines added
5.300 lines removed
1.875 lines modified

Every day

Kernel releases 4.0.0 – 4.5.0
April 2015 – March 2016

7,8 changes per hour

Kernel releases 4.0.0 – 4.5.0
April 2015 – March 2016

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.

8,1 changes per hour

4.3 release

This past 3.16 release was the fastest we have ever created. That number shows just how well the Linux kernel development model is working. We are growing in developers and in how fast we are developing overall.

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.

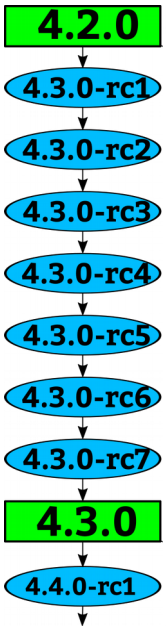
How we stay sane

Time based releases
Incremental changes

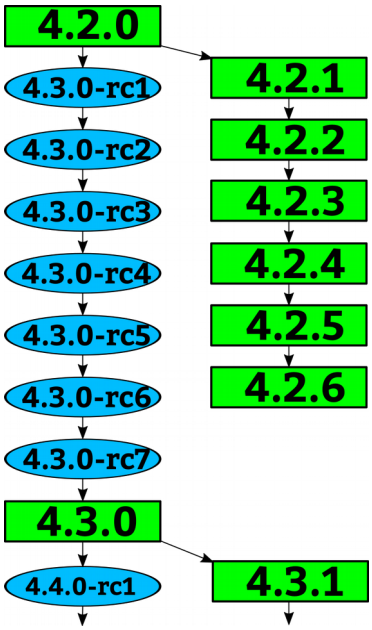
New release every 2½ months

Kernel releases 3.10.0 – 4.5.0

67 days to be exact, very regular experience.



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...

“Longterm kernels”

One picked per year
Maintained for two years

3.14 4.1 4.4

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.

3.10 and 3.14 are the longterm kernel releases I am maintaining.

3.10 will stop being maintained in October.

Ben Hutchings is maintaining the 3.2 kernel as a longterm kernel for the Debian project.

The LTSI project is based on the longterm kernels.

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

Kernel Security

Averaging 10 fixes per day.

If you look at the number of patches flowing into the stable tree, we are averaging 10 patches a day, every single day.

Now not all of them are “security” fixes. But some small percentage is.

This is for the latest kernel release, a 2 year old kernel, 3.14 is averaging 1/3 of all fixes apply to it.

The 3.2 kernel is averaging about 2 fixes a day.

> “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.

kernel.org sysadmin, in charge of the LF sysadmin team, Fedora infrastructure developer.

Great presentation on how you, as a sysadmin, can implement secure practices for your network. Full checklist and guide has been published.

But, even with those practices, we need low-level changes in order to save ourselves from the accidents that will happen.

We need “airbags” in the kernel, and elsewhere.

Things like SELinux, grsec, openwall we need them.

“We will always have bugs,
we must stop their exploitation”
– Kees Cook

outflux.net/slides/2015/ks/security.pdf

Kees Cook, kernel security developer, presentation at kernel summit last year.

We need to start doing things to make the kernel more “robust” from a security standpoint.

Even if it makes things harder for the developers.

Everyone agreed.

Kernel Hardening

kernsec.org/wiki/index.php/Kernel_Self_Protection_Project

Core Infrastructure Initiative

Kernel hardening project.

New security features are being added in each release, but if you don’t upgrade, you don’t get those features, and protection.

CII is helping to fund this, if you want to work on it, we need developers, and we will pay for it.

“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.

github.com/gregkh/kernel-development

Obligatory Penguin Picture