LINUX FOUNDATION

Linux Kernel Community

Greg Kroah-Hartman
gregkh@linuxfoundation.org

44,600 files
17,730,000 lines

3,061 developers
433 companies

Kernel releases 3.8.0 – 3.12.0
October 2012 – November 2013

8,600 lines added
3,900 lines removed
2,000 lines modified

Kernel releases 3.8.0 – 3.12.0
October 2012 – November 2013

8,600 lines added
3,900 lines removed
2,000 lines modified

Every Day

Kernel releases 3.8.0 – 3.12.0
October 2012 – November 2013

7.59 changes per hour

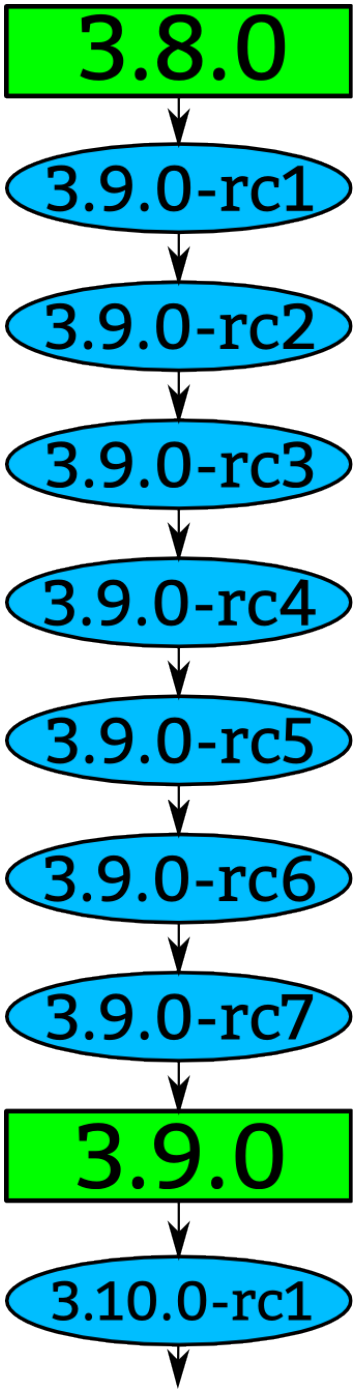
Kernel releases 3.8.0 – 3.12.0
October 2012 – November 2013

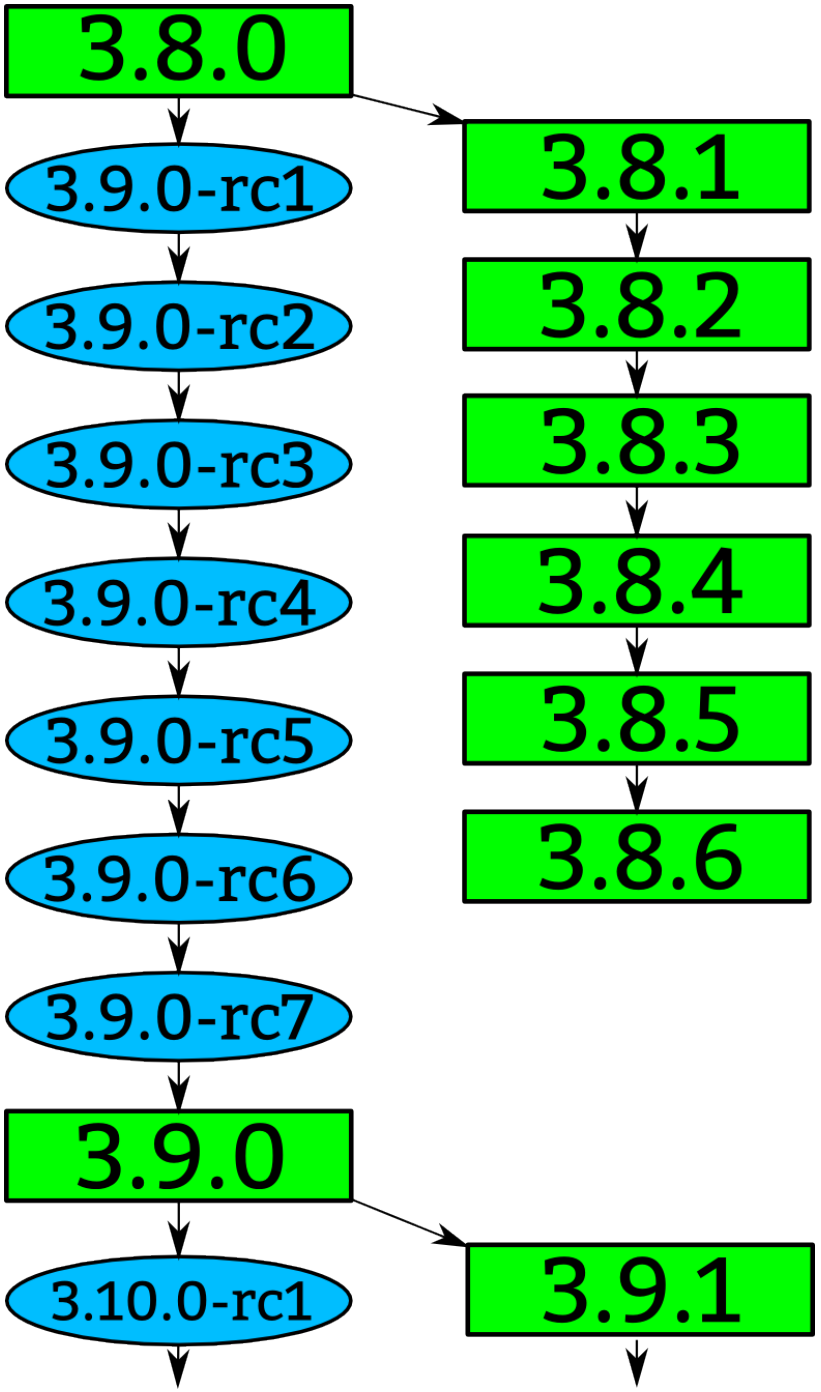
9.02 changes per hour

3.10.0 release

How we stay sane

Time based releases
Incremental changes


3.8.0
3.9.0-rc1
3.9.0-rc2
3.9.0-rc3
3.9.0-rc4
3.9.0-rc5
3.9.0-rc6
3.9.0-rc7
3.9.0
3.10.0-rc1

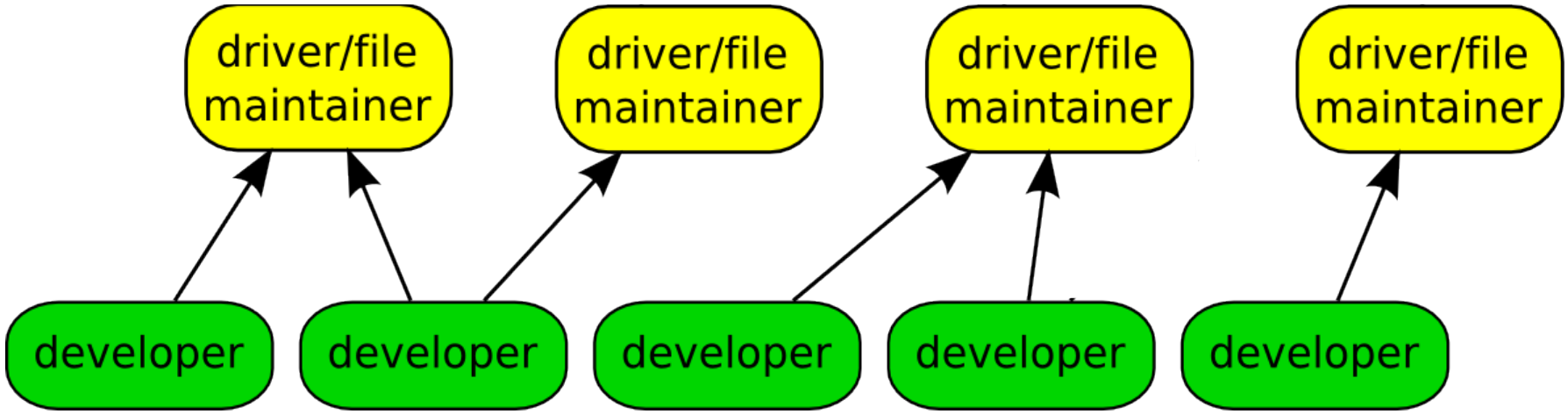

3.8.0
3.9.0-rc1
3.9.0-rc2
3.9.0-rc3
3.9.0-rc4
3.9.0-rc5
3.9.0-rc6
3.9.0-rc7
3.9.0
3.10.0-rc1
3.8.1
3.8.2
3.8.3
3.8.4
3.8.5
3.8.6
3.9.1

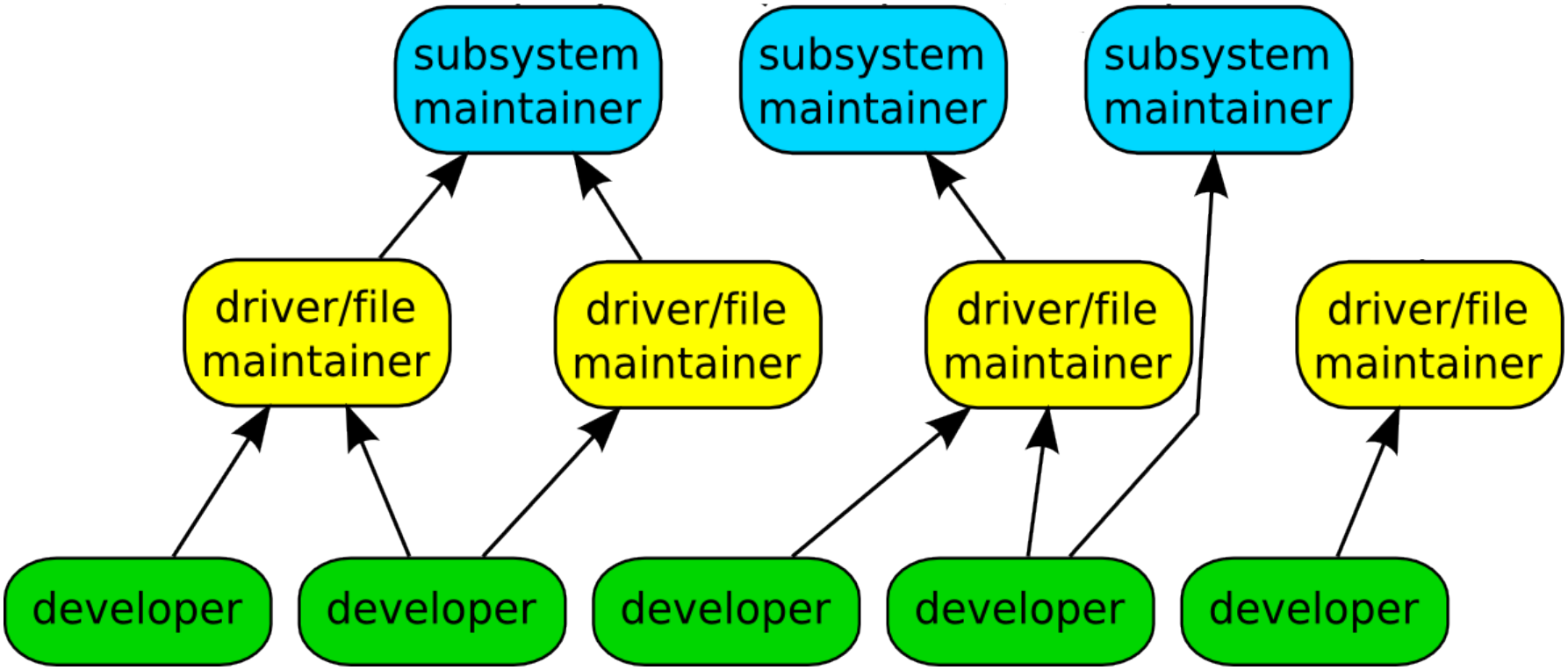
“Longterm kernels”

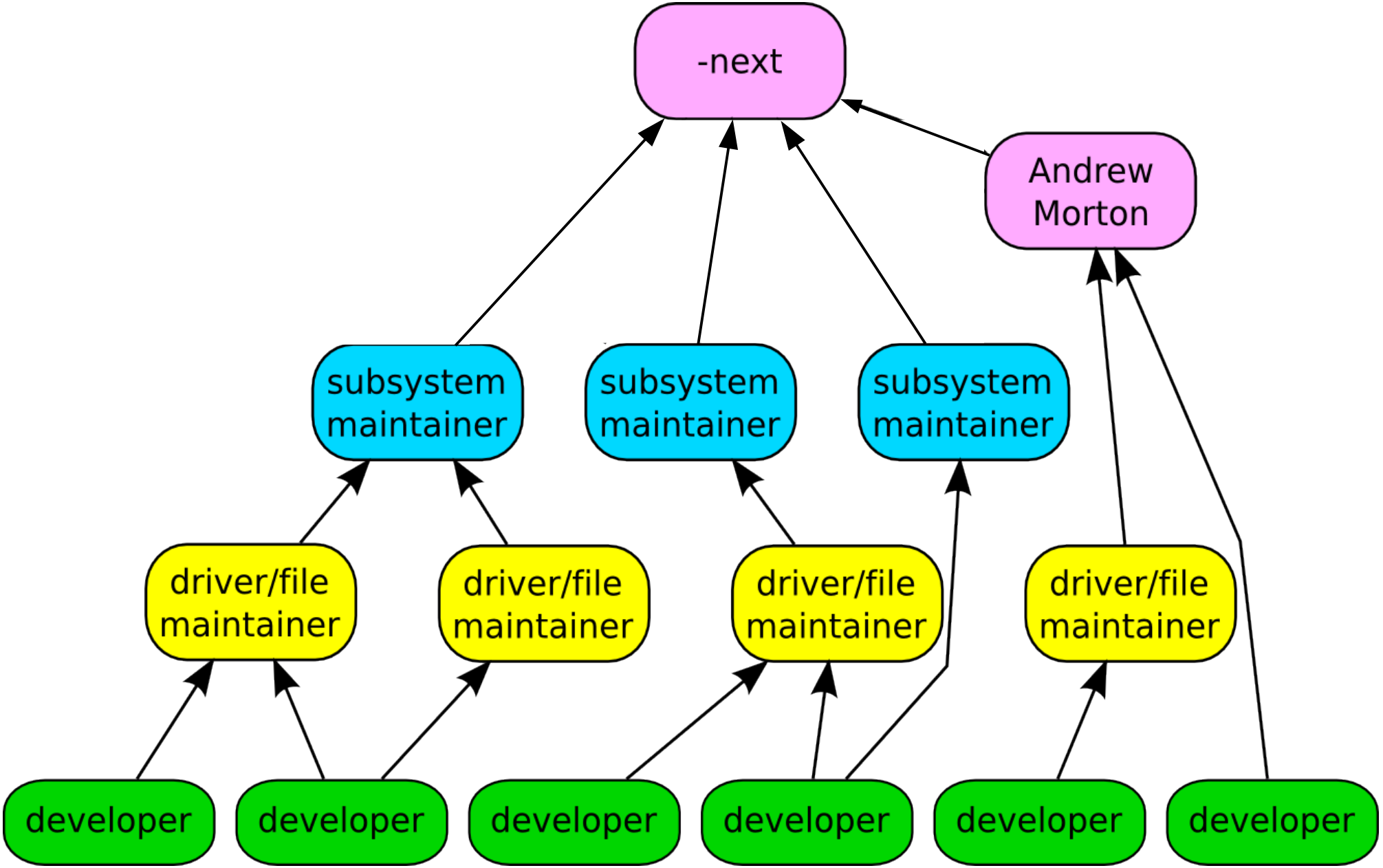
One picked per year
Maintained for two years

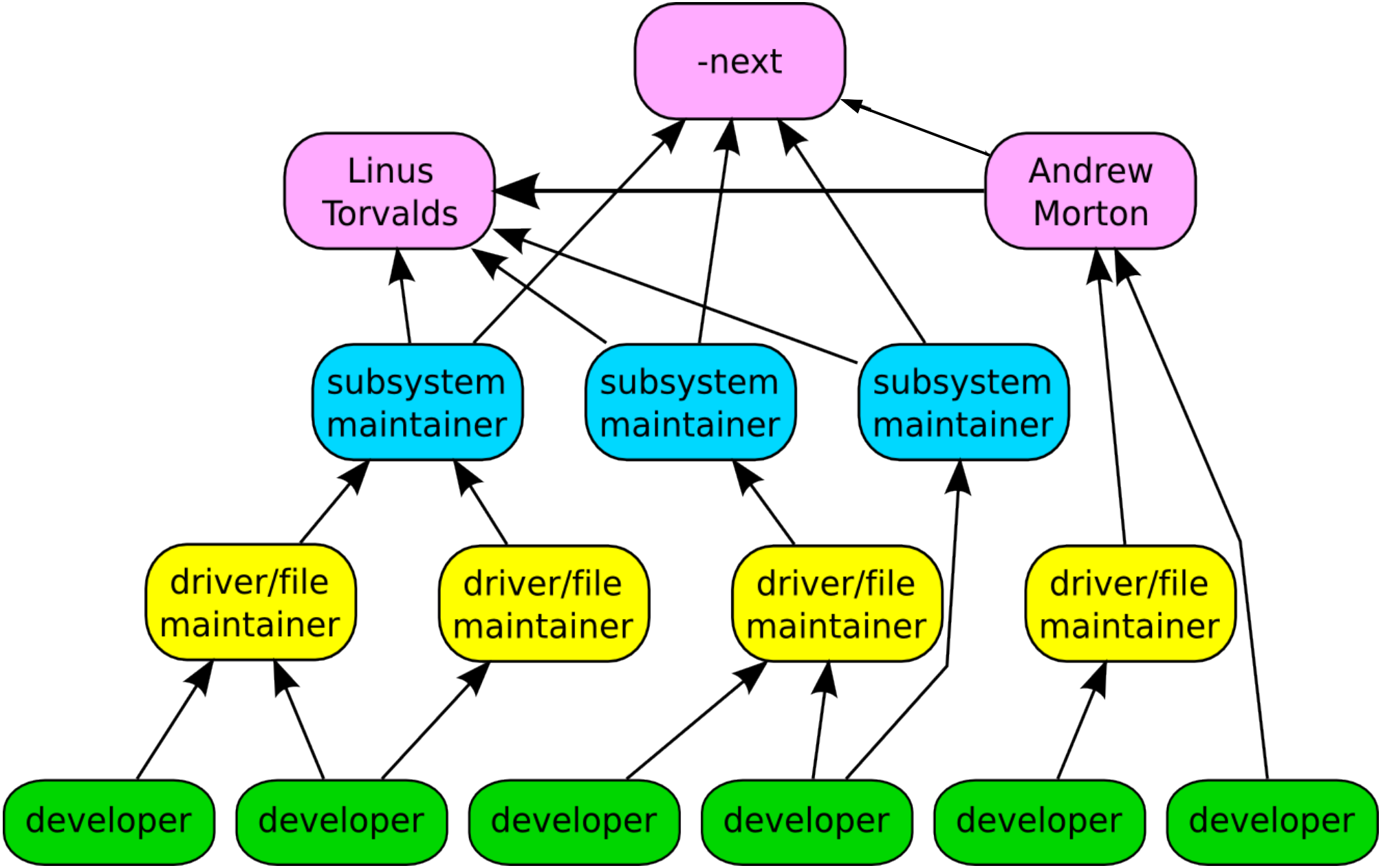
3.4 3.10


developer
developer
developer
developer
developer


driver/file maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
developer
developer
developer
developer
developer


subsystem maintainer
subsystem maintainer
subsystem maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
developer
developer
developer
developer
developer


-next
Andrew Morton
subsystem maintainer
subsystem maintainer
subsystem maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
developer
developer
developer
developer
developer
developer

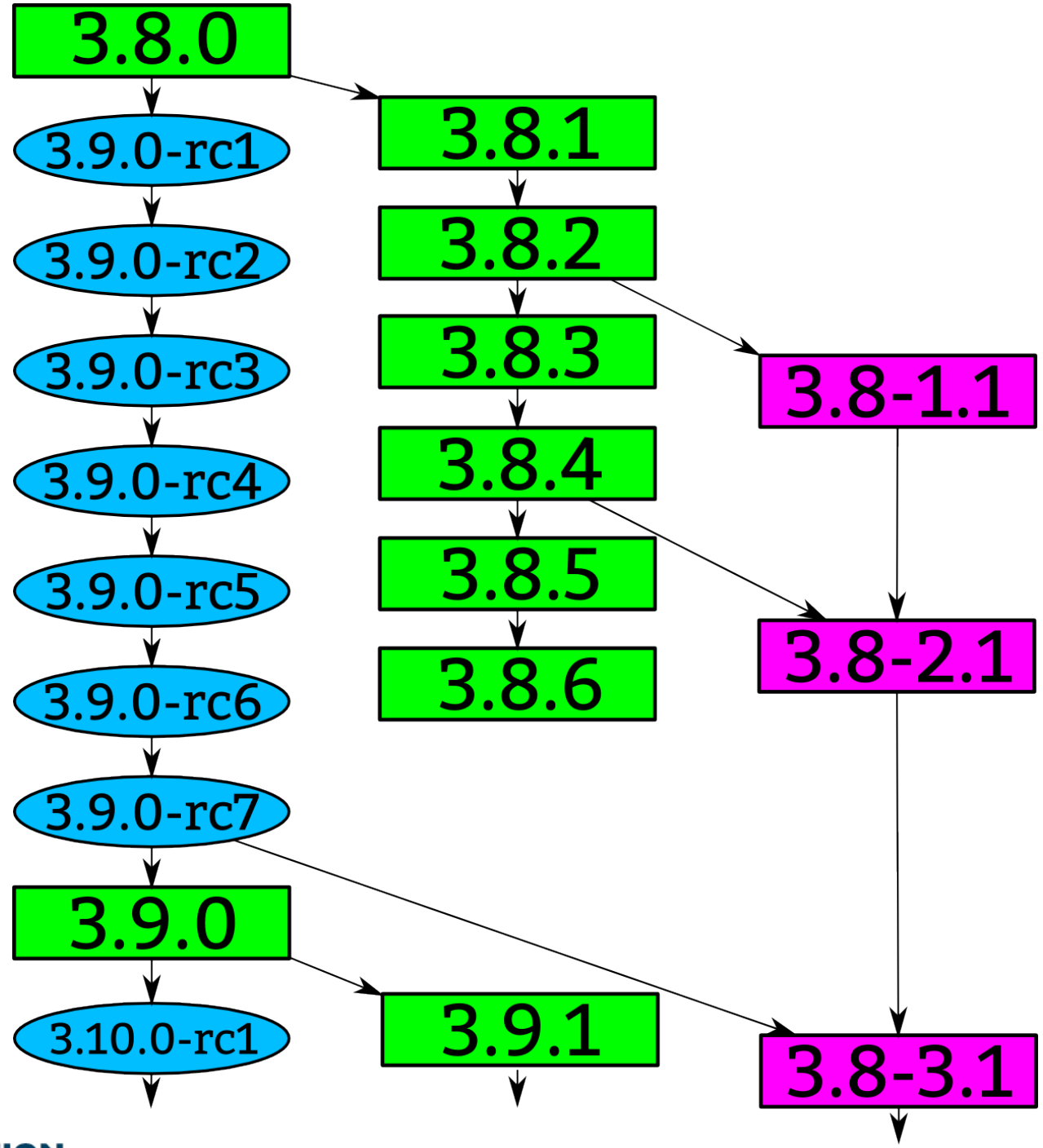

-next
Linus Torvalds
Andrew Morton
subsystem maintainer
subsystem maintainer
subsystem maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
developer
developer
developer
developer
developer
developer

Who is funding this work?

1. “Amateurs”	11.6%
2. Intel	9.3%
3. Red Hat	8.7%
4. Linaro	5.9%
5. Samsung	4.0%
6. Texas Instruments	3.7%
7. Unknown Individuals	3.6%
8. IBM	3.1%
9. SuSE	3.1%
10. Vision Engraving	2.7%

Who is funding this work?

11. Google	2.4%
12. Renesas	1.9%
13. Freescale	1.4%
14. Broadcom	1.3%
15. Oracle	1.3%
16. Consultants	1.3%
17. NVidia	1.2%
18. AMD	1.2%
19. LINBIT	1.1%
20. Free Electrons	1.1%


3.8.0
3.9.0-rc1
3.9.0-rc2
3.9.0-rc3
3.9.0-rc4
3.9.0-rc5
3.9.0-rc6
3.9.0-rc7
3.9.0
3.10.0-rc1
3.8.1
3.8.2
3.8.3
3.8.4
3.8.5
3.8.6
3.9.1
3.8-1.1
3.8-2.1
3.8-3.1

If your kernel provider is not a member of the community, your changes never get upstream.


©¡WOUW!

Linux Kernel Community

Greg Kroah-Hartman
gregkh@linuxfoundation.org

I'm going to discuss how fast the kernel is moving, how we do it, who does it, and how the enterprise distributions fit into the picture.

44,600 files
17,730,000 lines

Kernel release 3.12.0

This was for the 3.12 kernel release, which happened November 3, 2013.

3,061 developers
433 companies

Kernel releases 3.8.0 – 3.12.0
October 2012 – November 2013

This was the size of our community for the past year.

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

LINUX FOUNDATION

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

This is the first time we have had over 400 companies in a single year.

8,600 lines added
3,900 lines removed
2,000 lines modified

Kernel releases 3.8.0 – 3.12.0
October 2012 – November 2013

This is the rate of change we had for the past year.

8,600 lines added
3,900 lines removed
2,000 lines modified

Every Day

Kernel releases 3.8.0 – 3.12.0
October 2012 – November 2013

Every single day.

Crazy...

7.59 changes per hour

Kernel releases 3.8.0 – 3.12.0
October 2012 – November 2013

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

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

All of these changes are through the whole kernel.

For example, the core kernel is only 5% of the code, and 5% of the changes were to the core kernel. Drivers are 55% of the kernel, and 55% of the changes were on drivers. The change is completely proportional all across the kernel.

9.02 changes per hour

3.10.0 release

This past 3.10 release was the fastest we have ever created. That shows just how well the Linux kernel development model is working. We are growing in developers and in how fast we are developing overall.

This was just the patches that were accepted, not all of the patches that were submitted. Lots of patches are rejected and it usually takes a few times to get patches accepted.

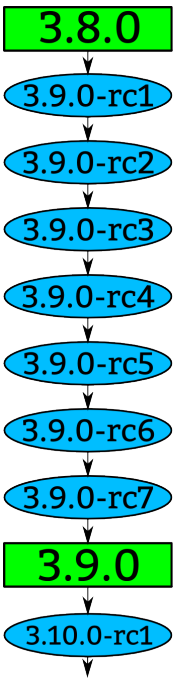
How we stay sane

Time based releases
Incremental changes

We do all of our releases based on time, not on features. We do a new release every 2 ½ months, and it doesn't matter what features are included in it.

This allows people to plan ahead as to what kernel version they are going to use in products, and it isn't a big deal if a feature misses a specific release, it can be included in the next one.

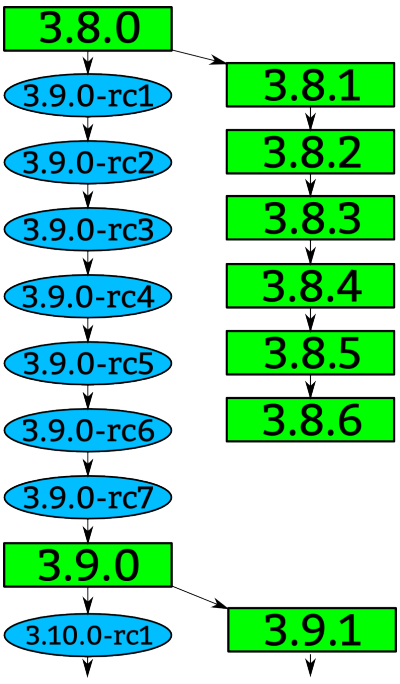
All of the changes in the kernel are small, incremental ones. We do not do large changes all at once. We break things down into smaller pieces, each one an individual step.



How a kernel is developed.

- Linus releases a stable kernel.
- 2 week merge window for subsystem maintainers
- rc1 is released (release candidate 1)
- Bugfixes only now
- 1 week later, rc2
- Bugfixes and regressions only.
- Another week, another rc.

And so on until all major bugfixes and regressions are resolved and then the cycle starts over again.



This worked well until we realized people wanted to use the kernel.org releases, and needed to include bugfixes as well.

So we created stable releases. Greg releases these every week or so. All of the fixes in these releases have to be in Linus's tree already, to prevent anything diverging.

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

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

“Longterm kernels”

One picked per year
Maintained for two years

3.4 3.10

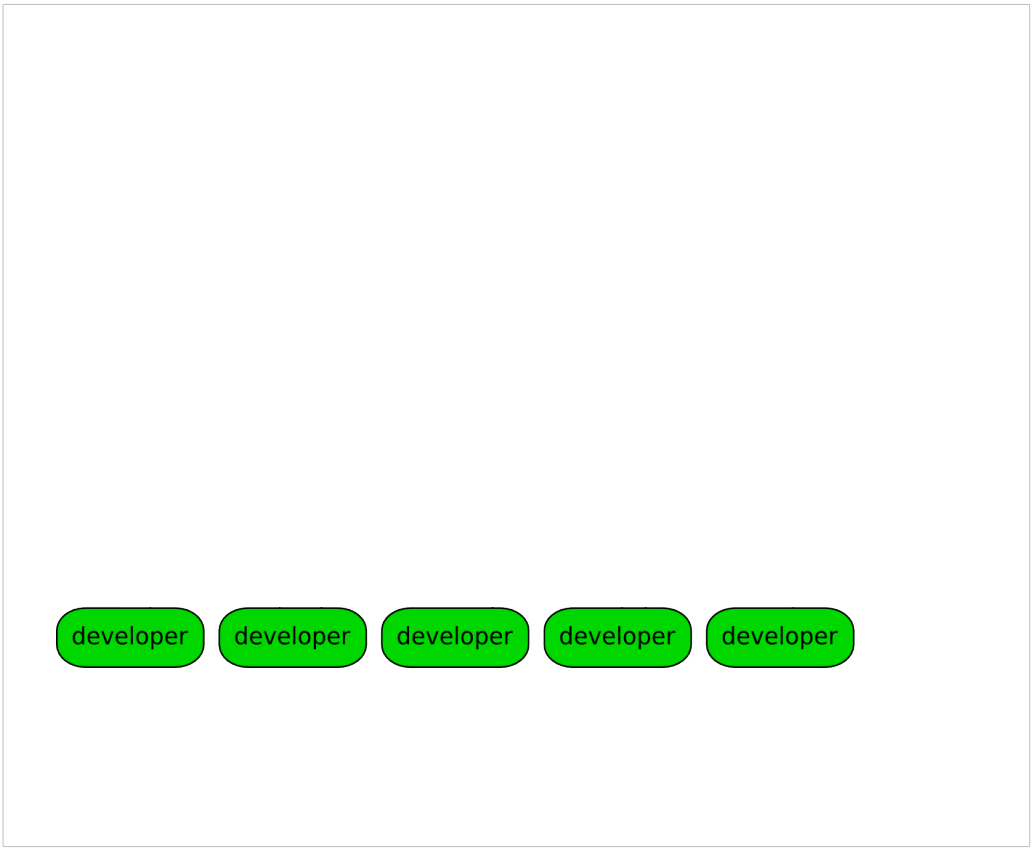
I pick one kernel release per year to maintain for 2 years.

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

3.4 and 3.10 are the longterm kernels that are currently being maintained.

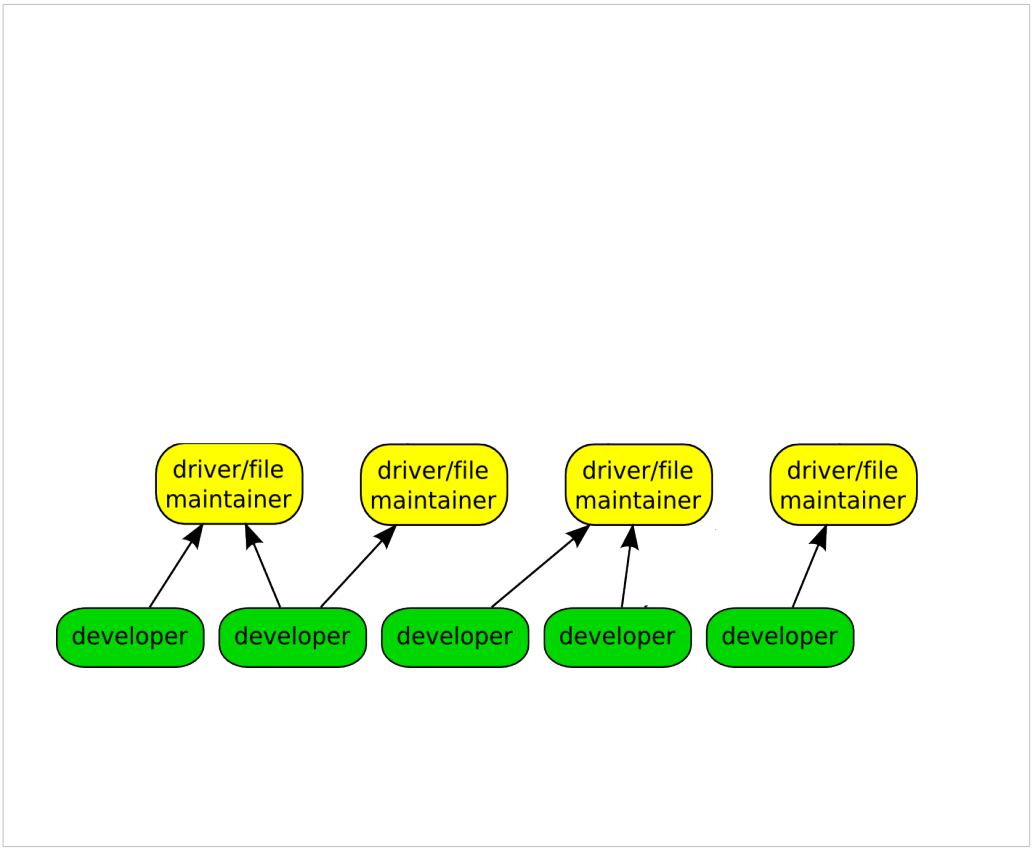
Ben Hutchings is also maintaining the 3.2 kernel as a longerm kernel for the Debian project.

The LTSI project is based on these longterm kernels.



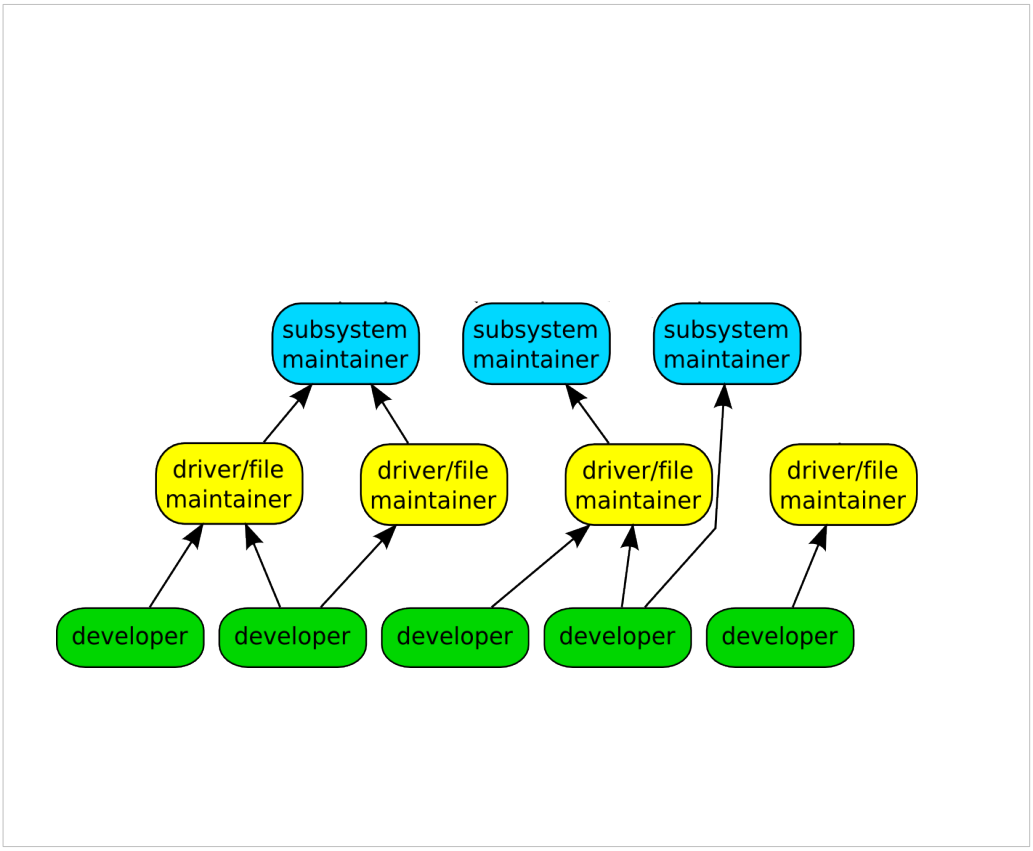
Like mentioned before, we have 3000 individual contributors. They all create a patch, a single change to the Linux kernel. This change could be something small, like a spelling correction, or something larger, like a whole new driver.

Every patch that is created only does one thing, and it can not break the build, complex changes to the kernel get broken up into smaller pieces.



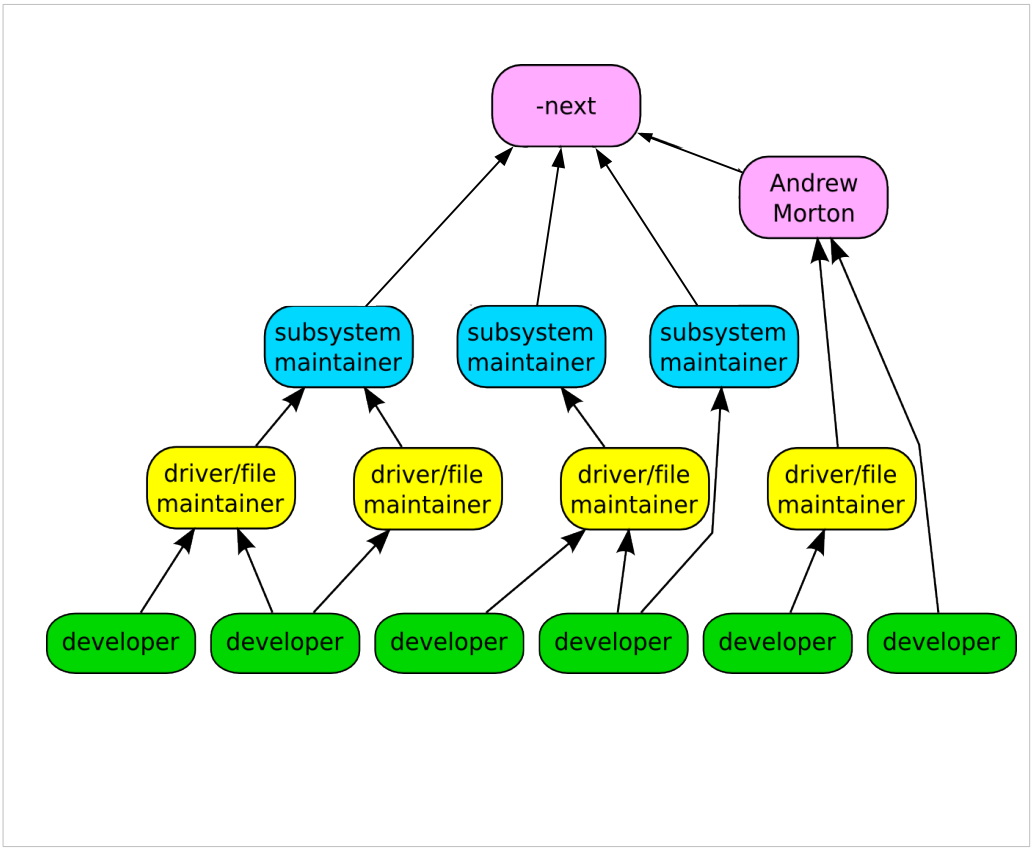
The developers send their patches off to the maintainer of the file or driver that they have modified.

We have about 700 different of these maintainers, all of them are listed in the MAINTAINERS file in the kernel source tree.



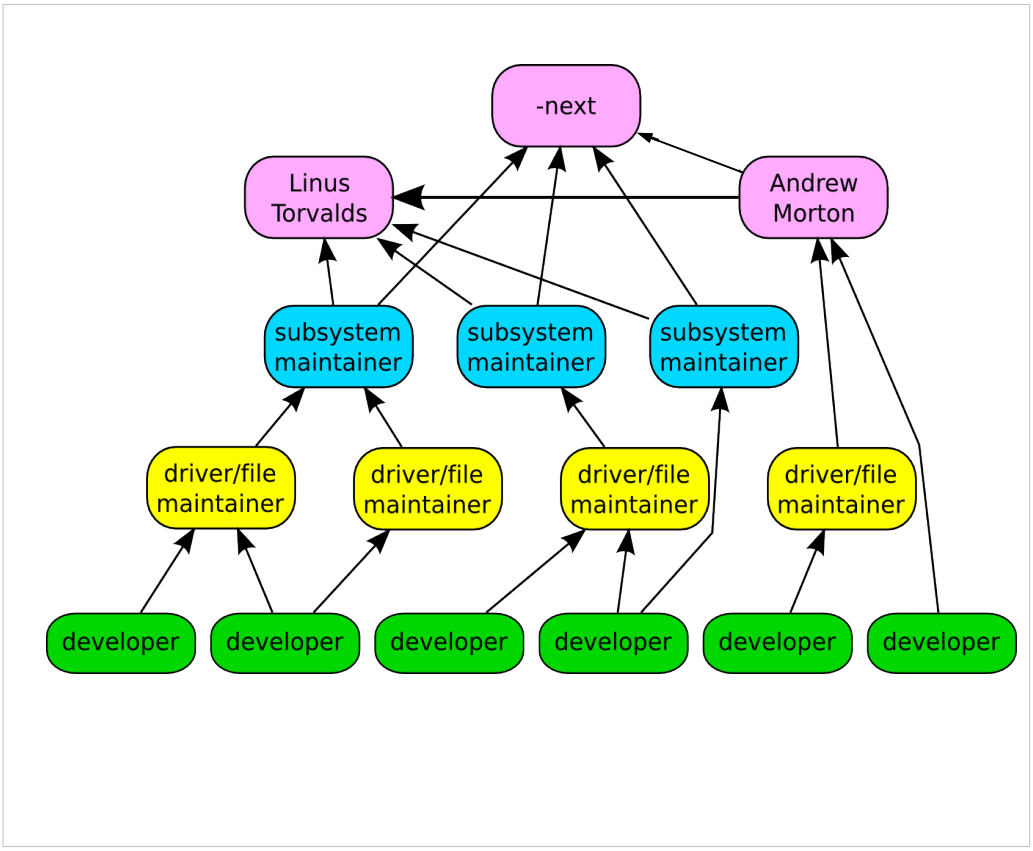
After reviewing the code, these maintainers send the patch off to the subsystem maintainer who is responsible for that portion of the kernel.

We have about 150 different subsystem maintainers.



Linux-next gets created every night from all of the different subsystm trees, and build tested on a wide range of different platforms.

Andrew Morton picks up patches that cross subsystems, or are missed by others, or are in areas where there are no maintainers.



Every 3 months, when the merge window opens up, everything gets sent to Linus from the subsystem maintainers.

The merge window is 2 weeks long, and thousands of patches get merged in that short time.

All of the patches merged to Linus should have been in the linux-next release.

Linus can't just pull directly from linux-next as sometimes the patches in the subsystem trees are just not ready to be merged, and it's up to the individual maintainer to decide what to send to Linus.

Who is funding this work?

1. “Amateurs”	11.6%
2. Intel	9.3%
3. Red Hat	8.7%
4. Linaro	5.9%
5. Samsung	4.0%
6. Texas Instruments	3.7%
7. Unknown Individuals	3.6%
8. IBM	3.1%
9. SuSE	3.1%
10. Vision Engraving	2.7%

Kernel releases 3.8.0 – 3.12.0

You can view this as 15% being done by “amateurs”, or 85% being done sponsored by companies.

Anyone who has shown any skill in doing kernel development is instantly hired, so very few people stay “amateurs” for very long.

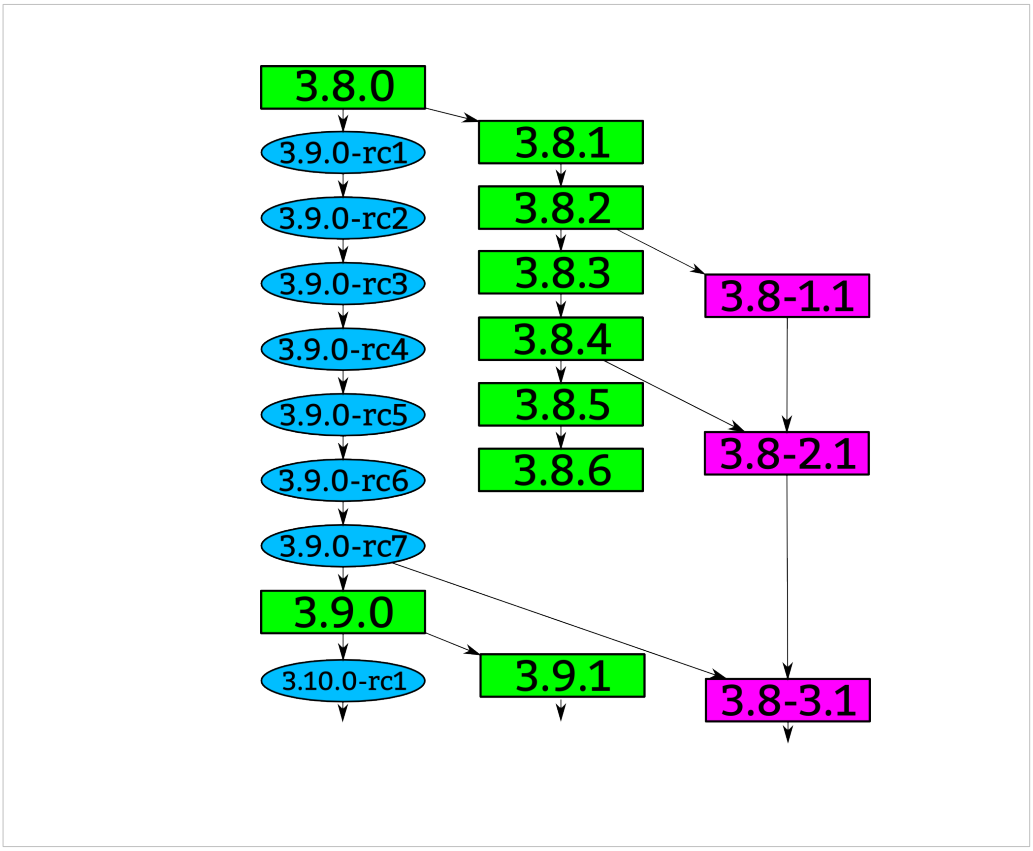
Vision Engraving is just 1 developer, contributing about 2,000 patches last year. One good developer can put your company in the top 10, so why aren't more companies doing this?

Who is funding this work?

11. Google	2.4%
12. Renesas	1.9%
13. Freescale	1.4%
14. Broadcom	1.3%
15. Oracle	1.3%
16. Consultants	1.3%
17. NVidia	1.2%
18. AMD	1.2%
19. LINBIT	1.1%
20. Free Electrons	1.1%

Kernel releases 3.8.0 – 3.12.0

Why all of this matters, if your company relies on Linux, and it depends on the future of Linux for supporting your needs, then you either trust these other companies are developing Linux in ways that will benefit you, or you need to get involved to make sure Linux works properly for your needs.



Enterprise kernels are based off of stable kernel releases.

Then they pick up patches from other stable releases, and some -rc releases, and are maintained for a very long period of time (7-10 years.)

The kernels here differ greatly in places from upstream kernel.org releases, which is fine, as long as you have a team that can support them. Most of the major enterprise distros have such developers, and they make sure that the fixes they find are pushed back upstream into kernel.org releases.

If your kernel provider is not a member of the community, your changes never get upstream.

February 18, 2013

But, if your kernel provider is not a member of the community, which is almost always the case for all embedded companies, then the changes and fixes in your kernel never get upstream, and usually get dropped between releases, which you need to be aware of if you rely on these vendors.

Obligatory Penguin picture.