Linux Kernel Development

Greg Kroah-Hartman
gregkh@linuxfoudation.org

github.com/gregkh/kernel-development

38,566 files

15,384,000 lines

Kernel release 3.4.0

2,833 developers
373 companies

Kernel releases 3.0.0 – 3.4.0
May 2011 – May 2012

7.21 changes per hour

3.4.0 release

10,500 lines added

8,400 lines removed

2,300 lines modified

Kernel releases 3.0.0 – 3.4.0
May 2011 – May 2012

10,500 lines added

8,400 lines removed

2,300 lines modified

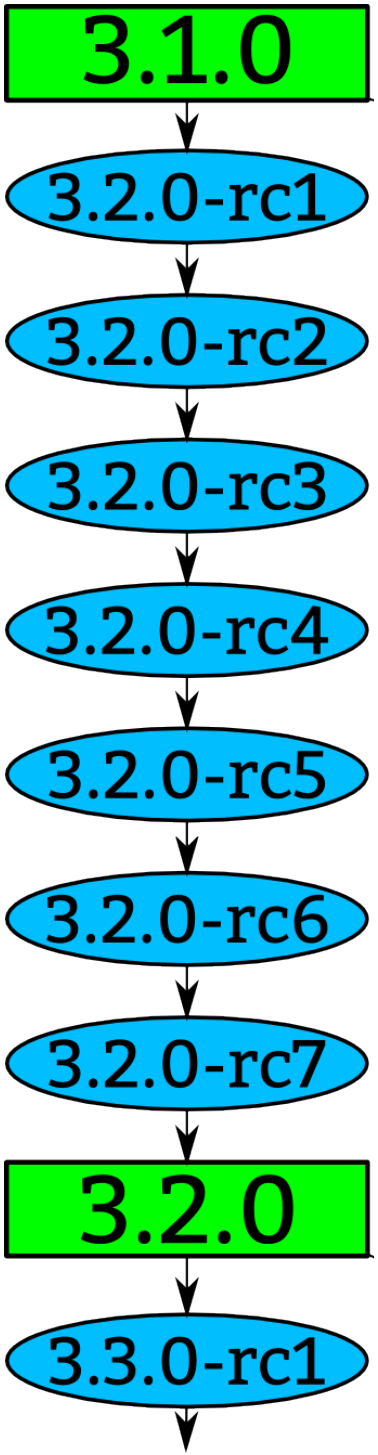
every day

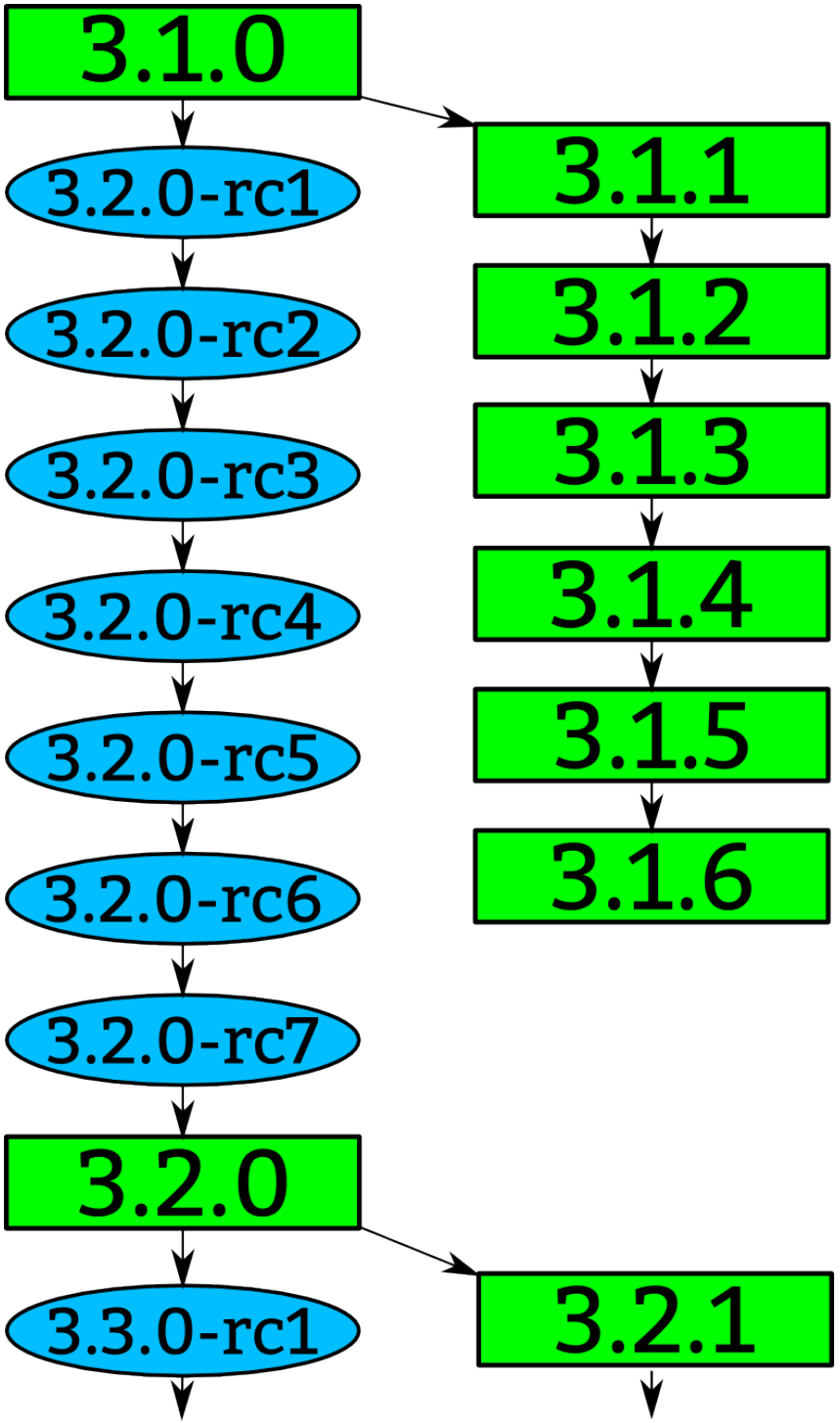
Kernel releases 3.0.0 – 3.4.0
May 2011 – May 2012

5.79 changes per hour

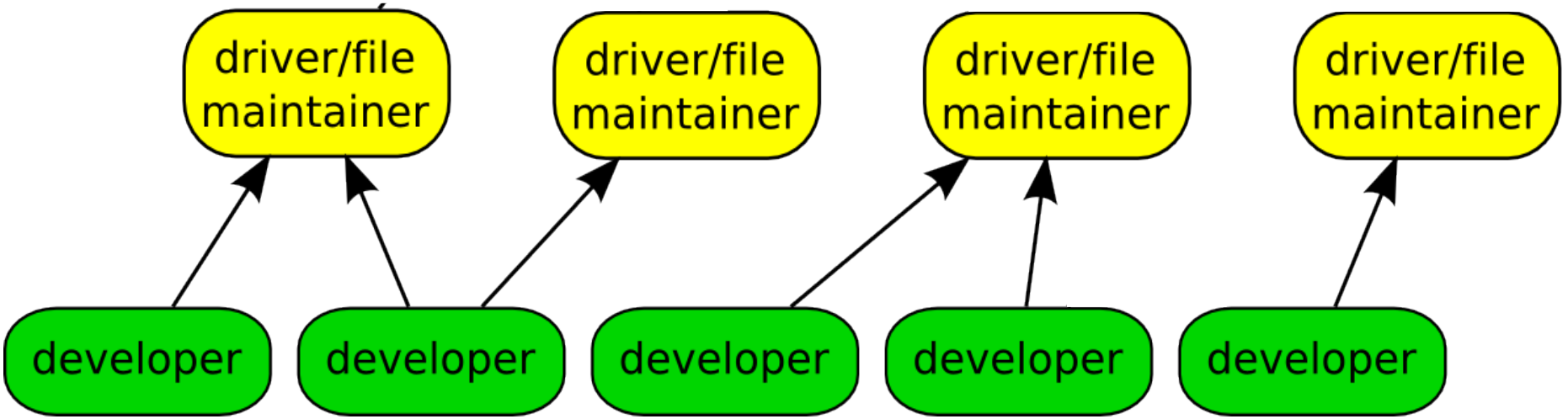
Kernel releases 3.0.0 – 3.4.0
May 2011 – May 2012

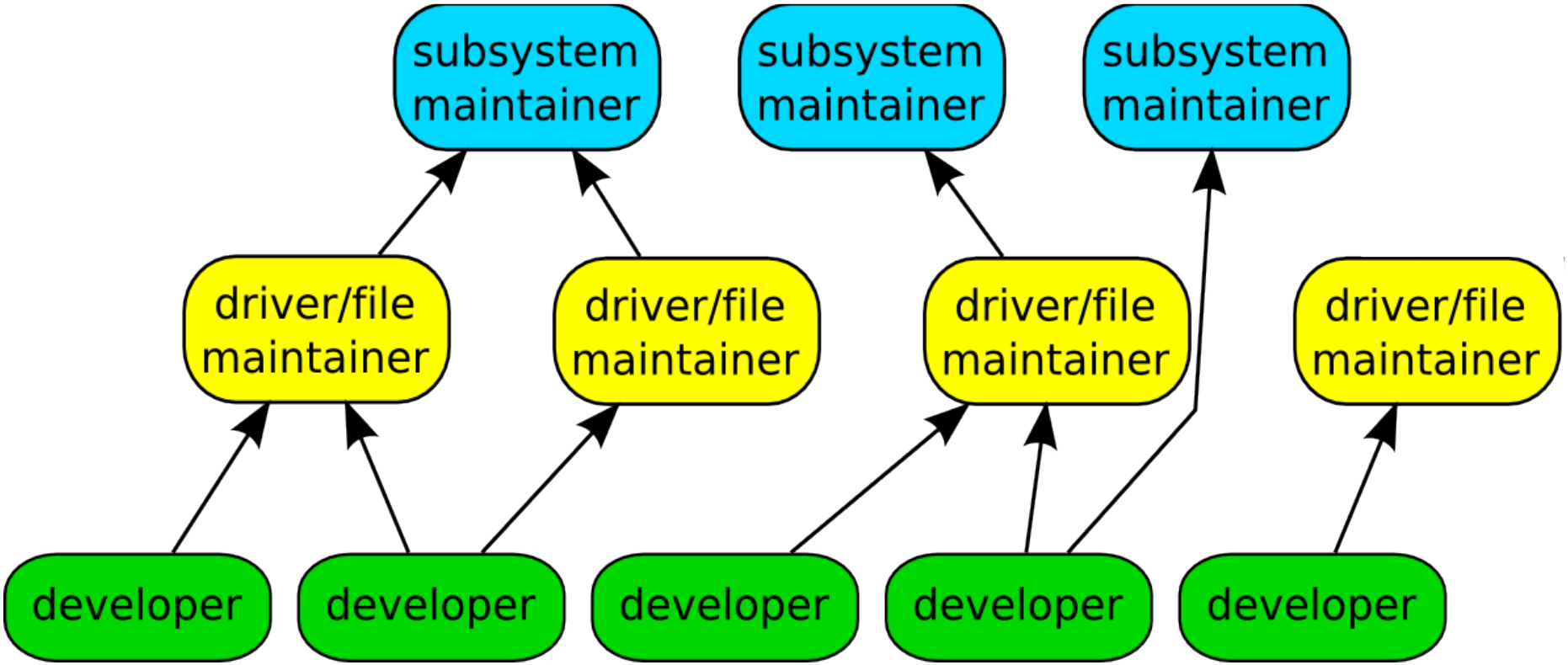

New release every
2¾ months
Kernel releases 2.6.20 – 3.4.0

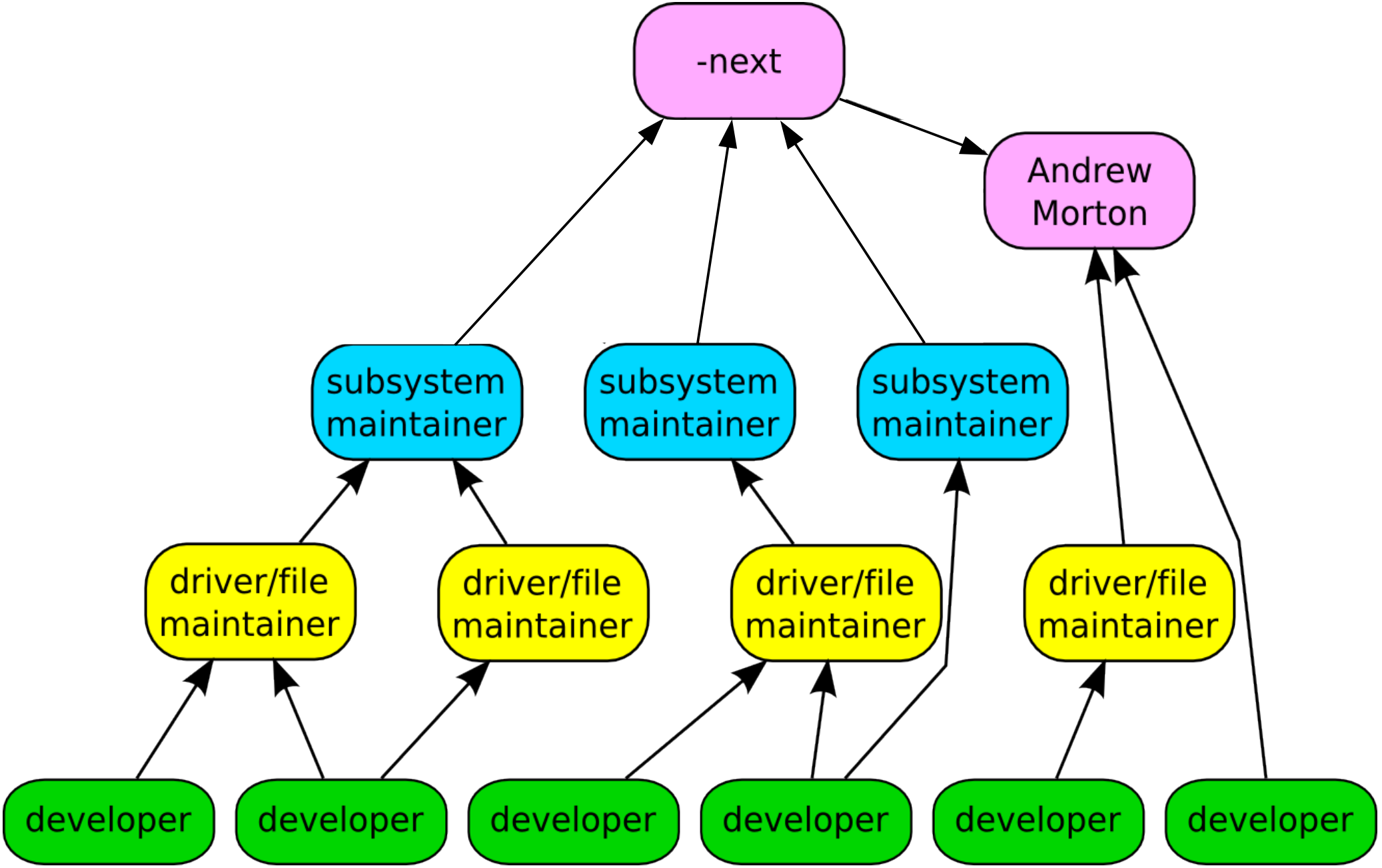

3.1.0
3.2.0-rc1
3.2.0-rc2
3.2.0-rc3
3.2.0-rc4
3.2.0-rc5
3.2.0-rc6
3.2.0-rc7
3.2.0
3.3.0-rc1

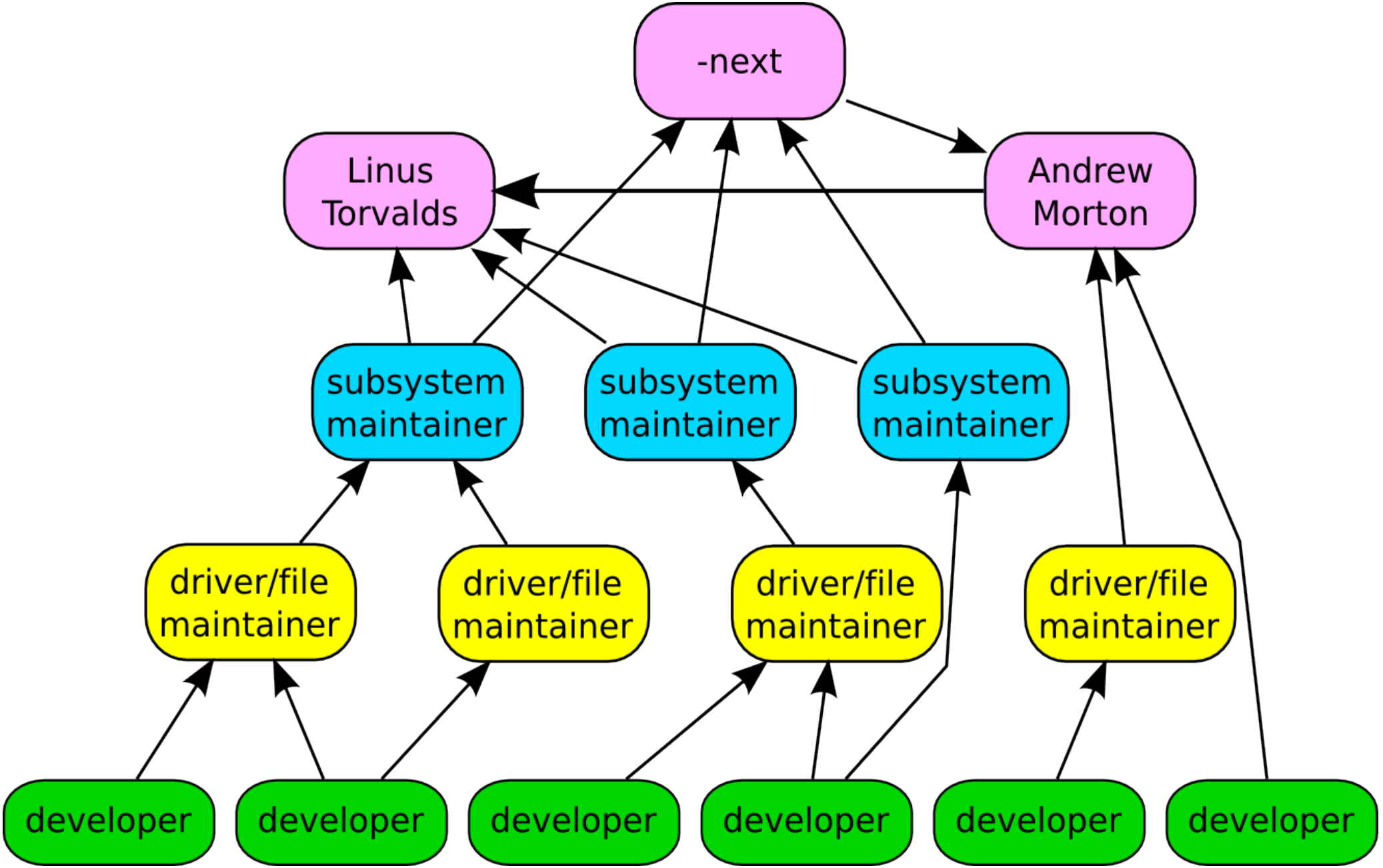

3.1.0
3.2.0-rc1
3.2.0-rc2
3.2.0-rc3
3.2.0-rc4
3.2.0-rc5
3.2.0-rc6
3.2.0-rc7
3.2.0
3.3.0-rc1
3.1.1
3.1.2
3.1.3
3.1.4
3.1.5
3.1.6
3.2.1


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

Top developers by quantity

Mark Brown	1026
Alxel Lin	723
K. Y. Srinivasan	626
Al Viro	607
Takashi Iwai	517
Mauro Chehab	507
Russell King	419
Johannes Berg	469
Ben Skeggs	405
Jonathan Cameron	396

Kernel releases 3.0.0 – 3.4.0

Top Signed-off-by:

Greg Kroah-Hartman	4767
David S. Miller	3857
John Linville	3252
Mauro Carvalho Chehab	2412
Mark Brown	2230
Linus Torvalds	1984
Andrew Morton	1573
James Bottomley	1089
Takashi Iwai	953
Russell King	930

Kernel releases 3.0.0 – 3.4.0

Who is funding this work?

1. “Amateurs”	14.2%
2. Red Hat	10.1%
3. Intel	8.6%
4. Unknown Individuals	5.2%
5. Novell	4.0%
6. IBM	3.7%
7. Texas Instruments	3.6%
8. Broadcom	3.0%
9. Consultants	2.3%
10. Wolfson Micro	2.1%

Kernel releases 3.0.0 – 3.4.0

Who is funding this work?

11. Samsung	1.9%
12. Google	1.8%
13. Oracle	1.7%
14. Freescale	1.5%
15. MiTAC	1.4%
16. Qualcomm	1.4%
17. Microsoft	1.3%
18. Linaro	1.2%
19. Nokia	1.2%
20. AMD	1.1%

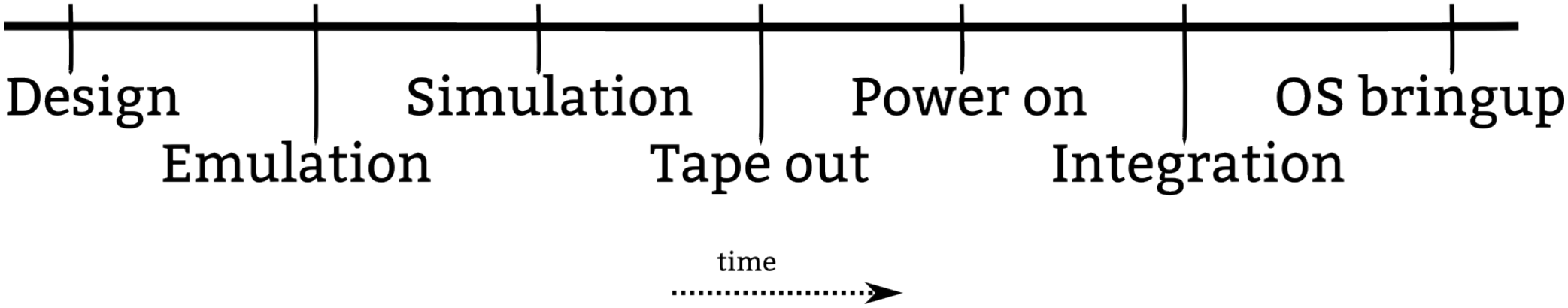
Kernel releases 3.0.0 – 3.4.0

Who is funding this work?

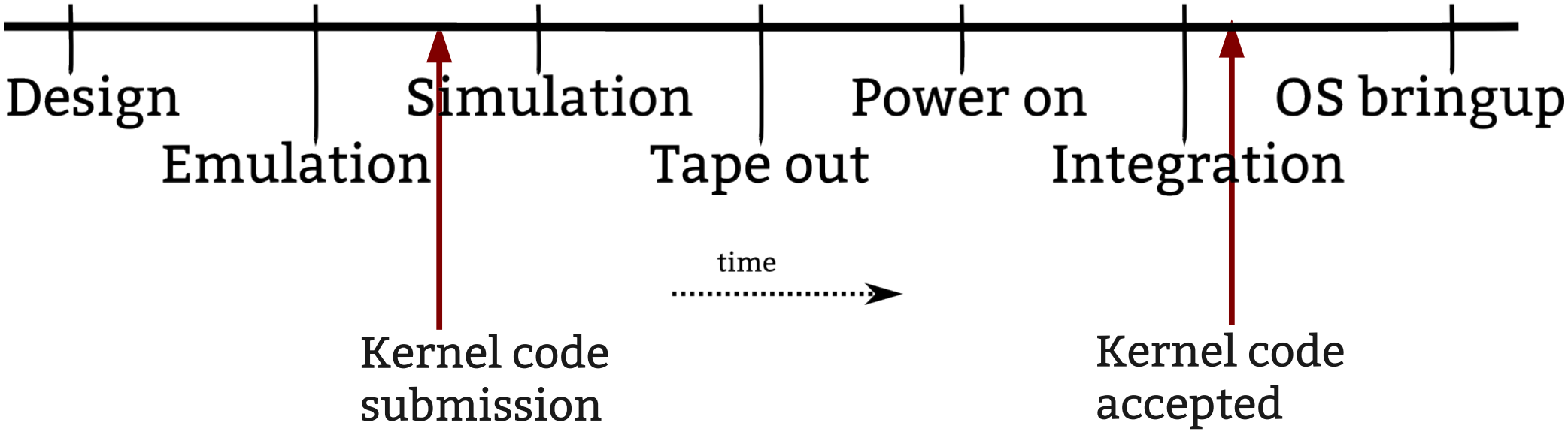
...	
23. Fujitsu	448
...	
43. Cisco	206
...	
59. HP	118
...	
230. Huawei	3

Kernel releases 3.0.0 – 3.4.0

Development Process



Development Process



Ideal

“Working upstream saves time and money”

Dan Frye – VP Open Systems, IBM
Dirk Hohndel – Chief Technologist, Intel

How to change

How to change

Submit code early and often

How to change

Send small pieces

How to change

Ask community for feedback

How to change

Act on it

How to change

Remove legal hurdles

How to change

Force them to work in public

How to change

Allow them to be the community

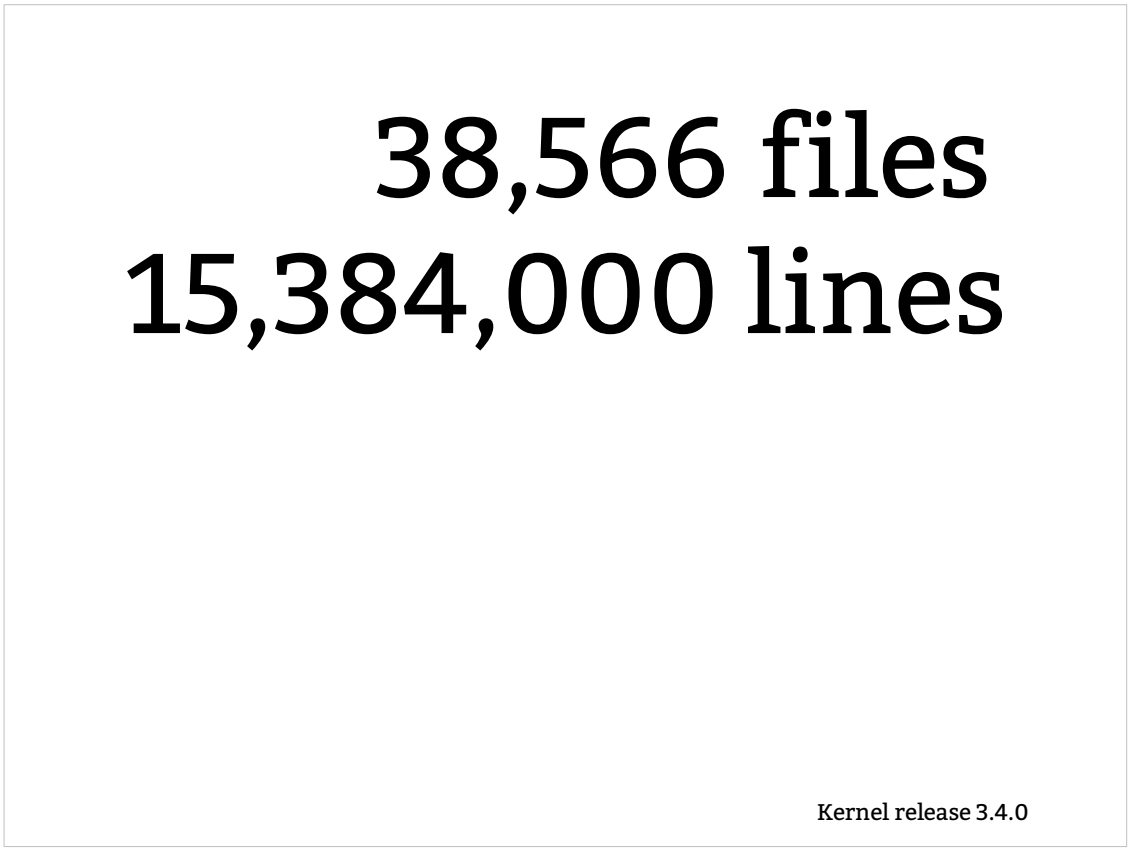
github.com/gregkh/kernel-development

Linux Kernel Development

Greg Kroah-Hartman
gregkh@linuxfoudation.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.



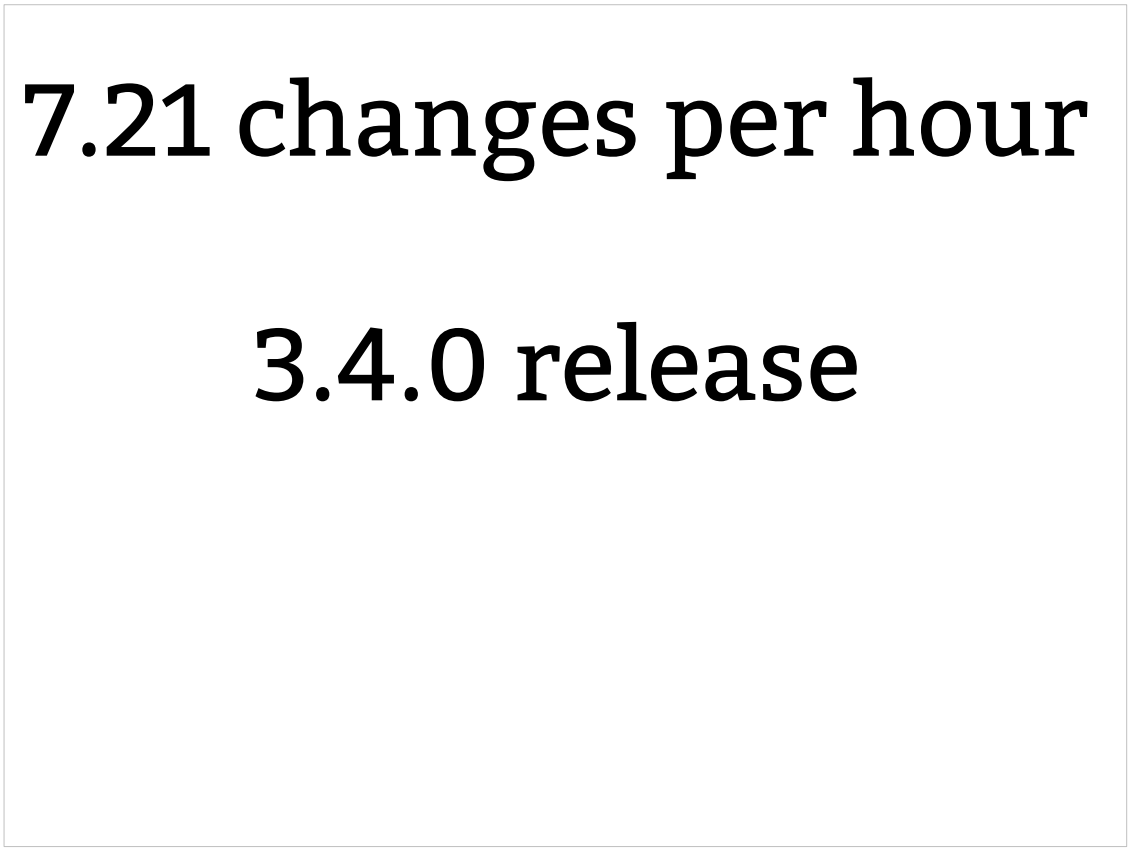
This was for the 3.4 kernel release, which happened May 20, 2012.

2,833 developers
373 companies

Kernel releases 3.0.0 – 3.4.0
May 2011 – May 2012

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.



The 3.4 kernel release was the fastest one ever created

10,500 lines added
8,400 lines removed
2,300 lines modified

Kernel releases 3.0.0 – 3.4.0
May 2011 – May 2012

10,500 lines added
8,400 lines removed
2,300 lines modified

every day

Kernel releases 3.0.0 – 3.4.0
May 2011 – May 2012

5.79 changes per hour

Kernel releases 3.0.0 – 3.4.0
May 2011 – May 2012

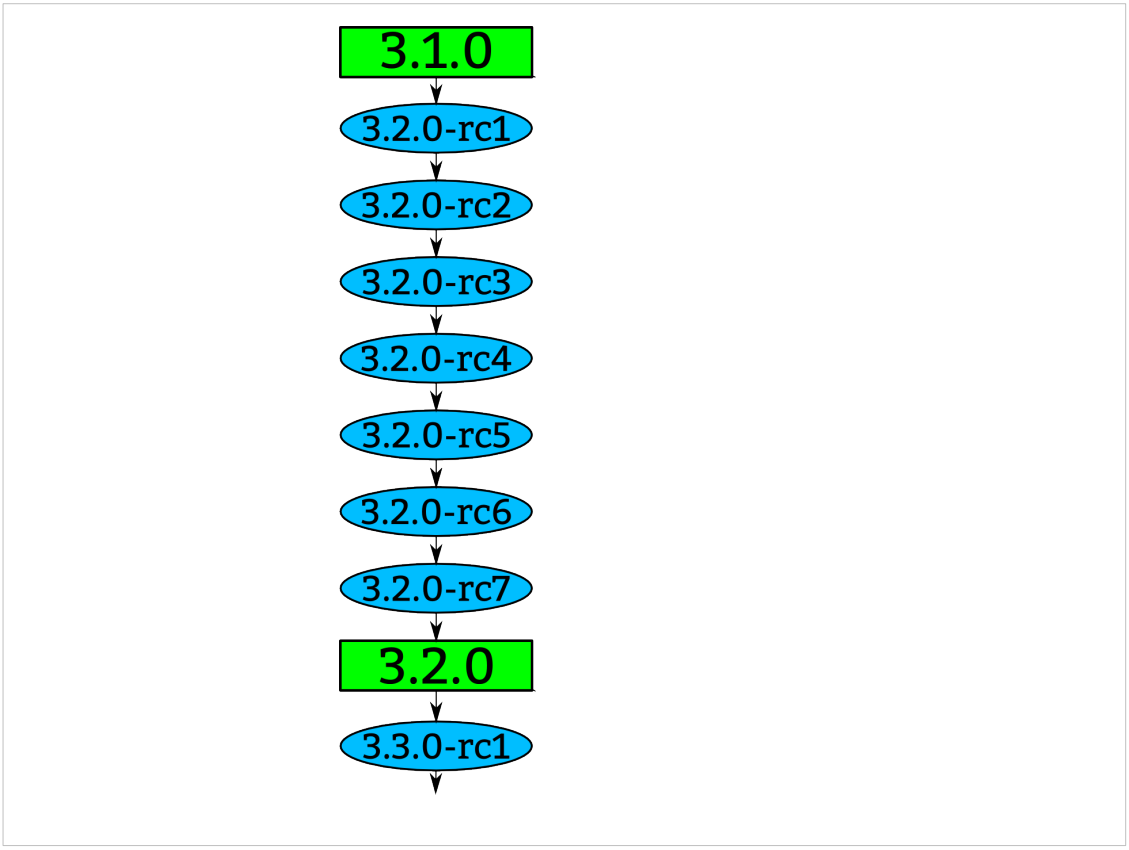
This is 24 hours a day, 7 days a week, for a full year.

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.

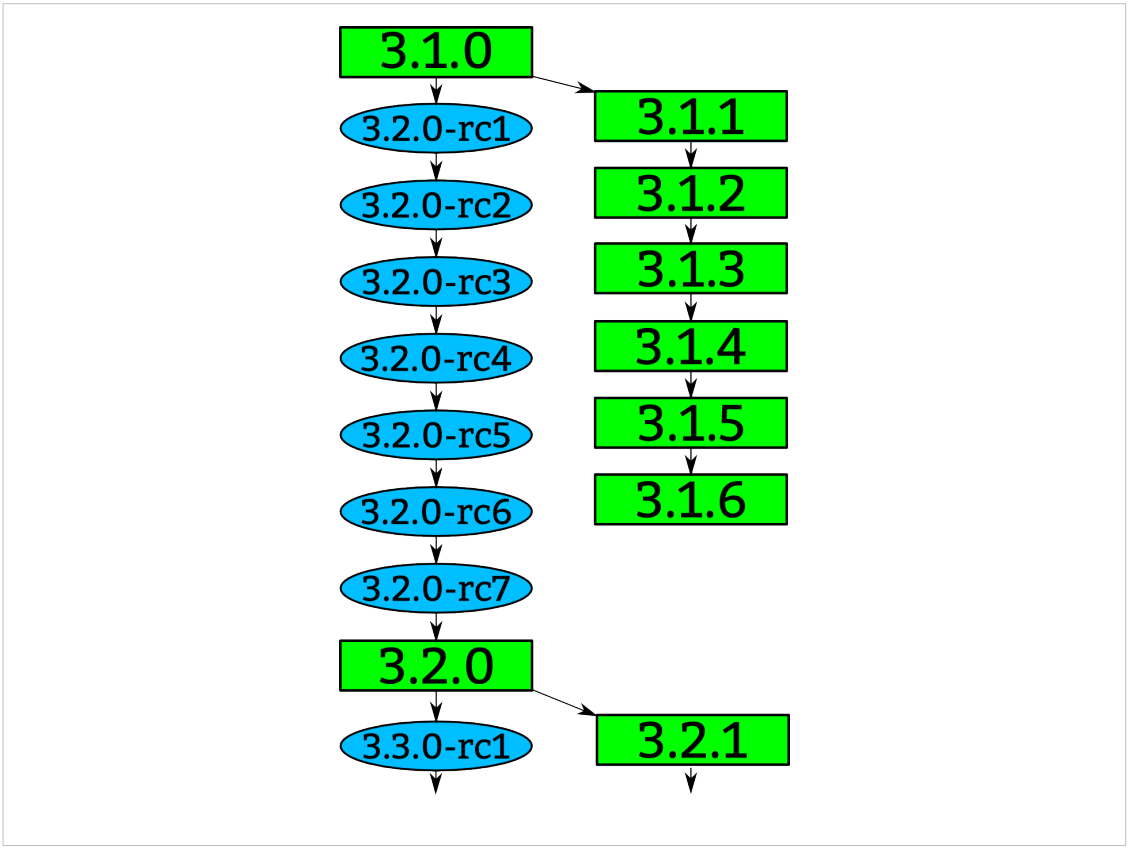
For example, the core kernel is only 5% of the code, and 5% of the change was to the core kernel. Drivers are 55%, and 55% was done to them, it's completely proportional all across the whole kernel.

84 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.

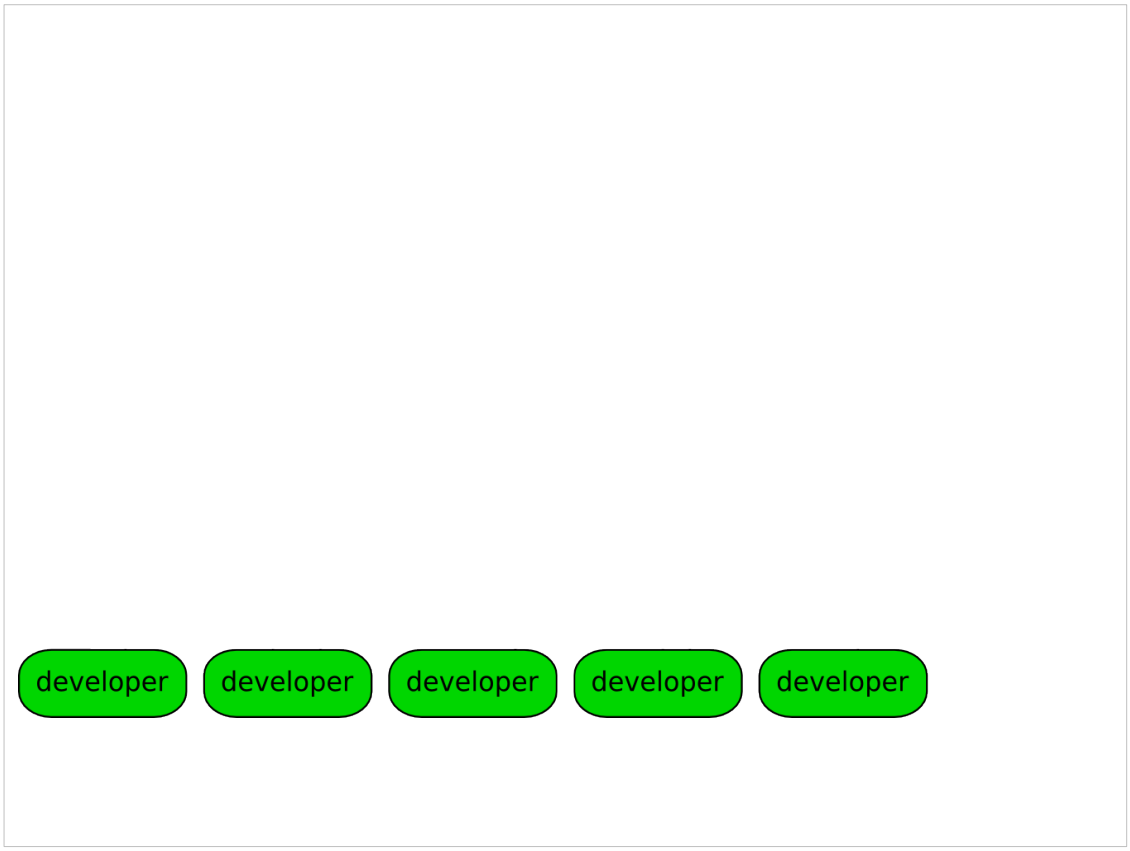


Greg takes the stable releases from Linus, and does stable releases with them, applying only fixes that are already in Linus's tree.

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

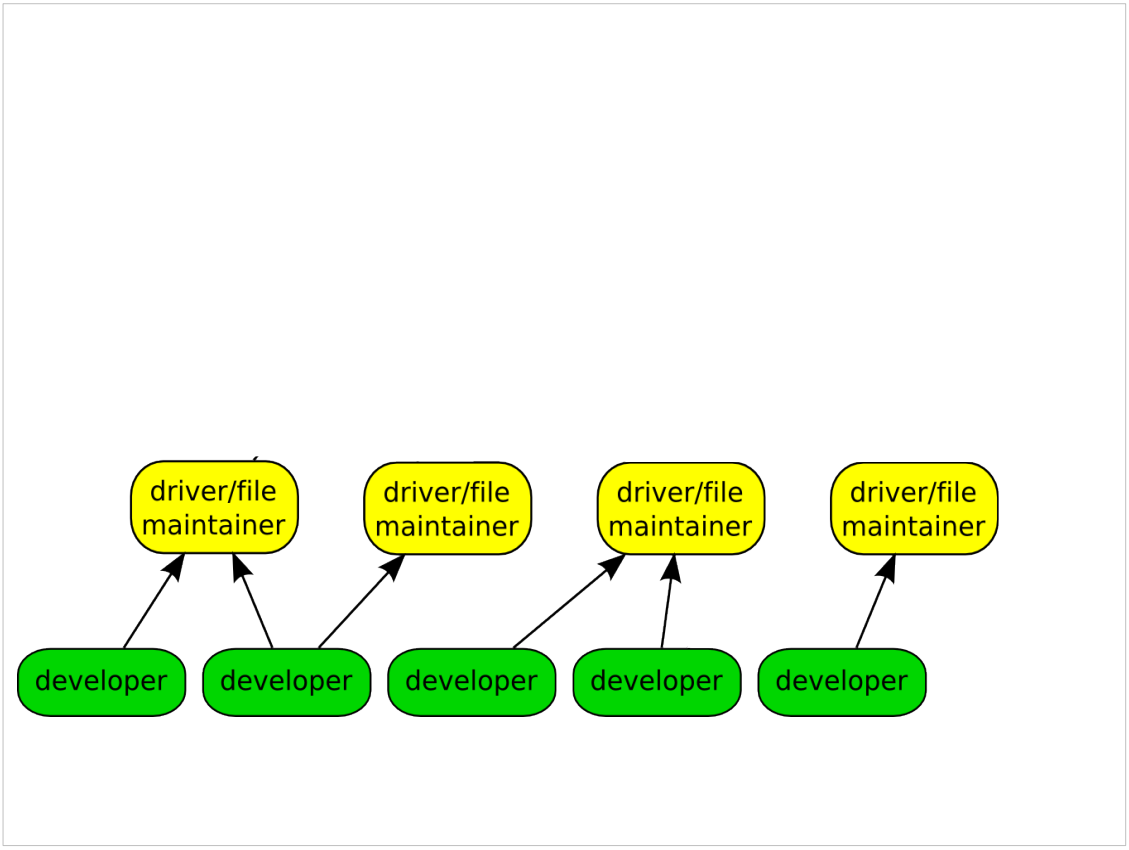
After Linus releases a new stable release, the old stable series is dropped.

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



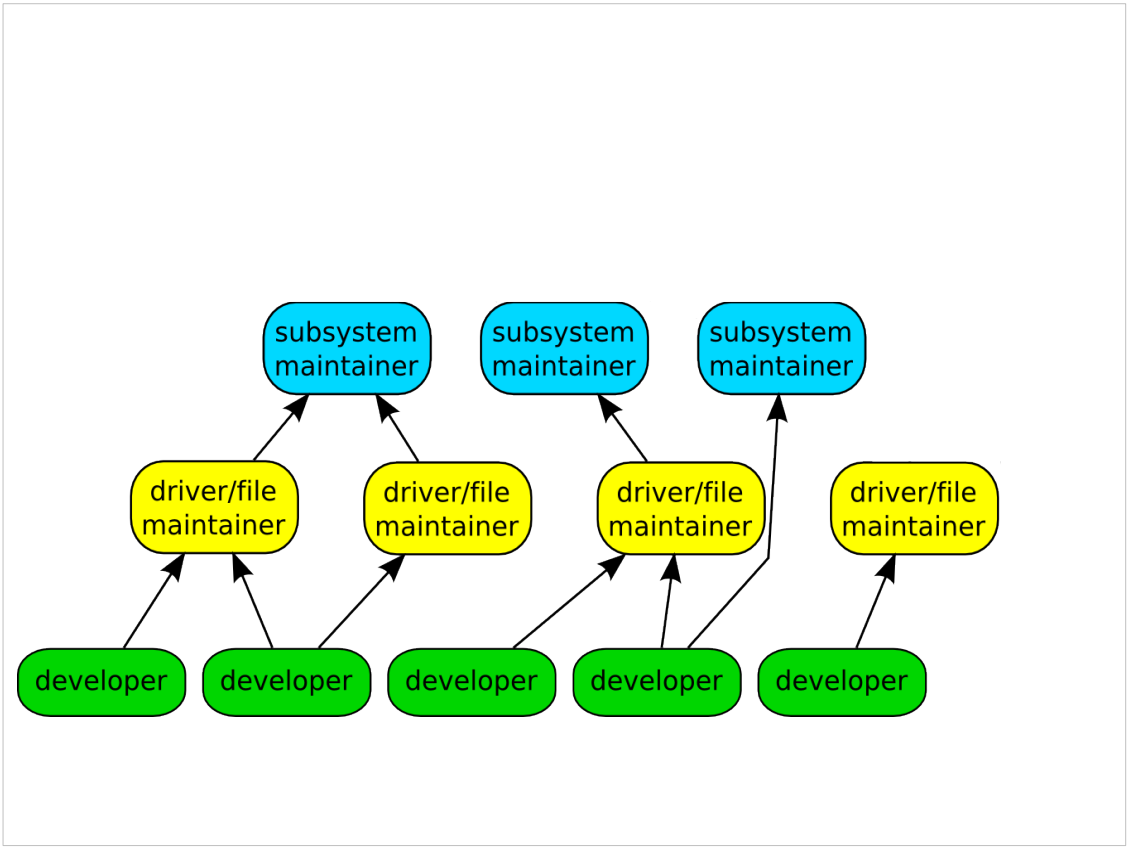
Like mentioned before, we have almost 2900 individual contributors. They all create a patch, a single change to the Linux kernel. This change could be something small, like a spelling correction, or something larger, like a whole new driver.

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.



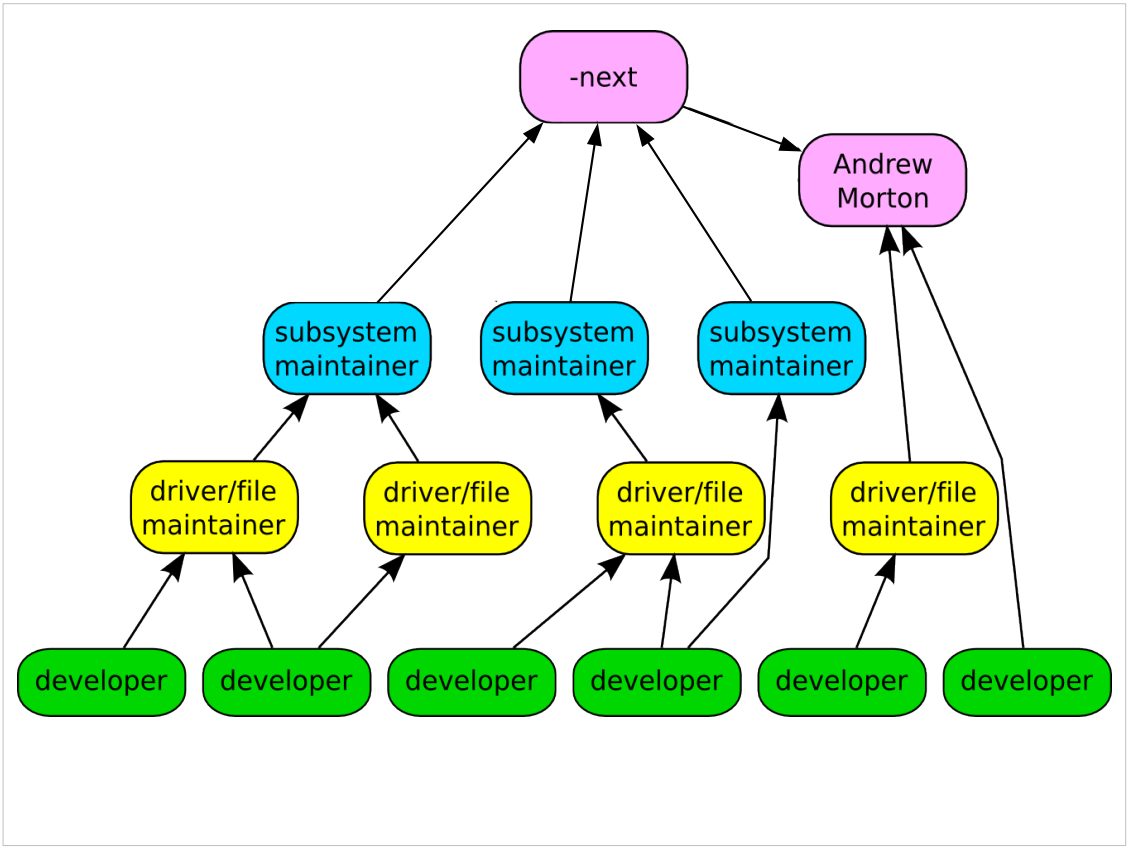
The developers send their patch to the maintainer of the file(s) that they have modified.

We have about 700 different driver/file/subsystem maintainers



After reviewing the code, and adding their own signed-off-by to the patch, the file/driver maintainer sends the patch to the subsystem maintainer responsible for that portion of the kernel.

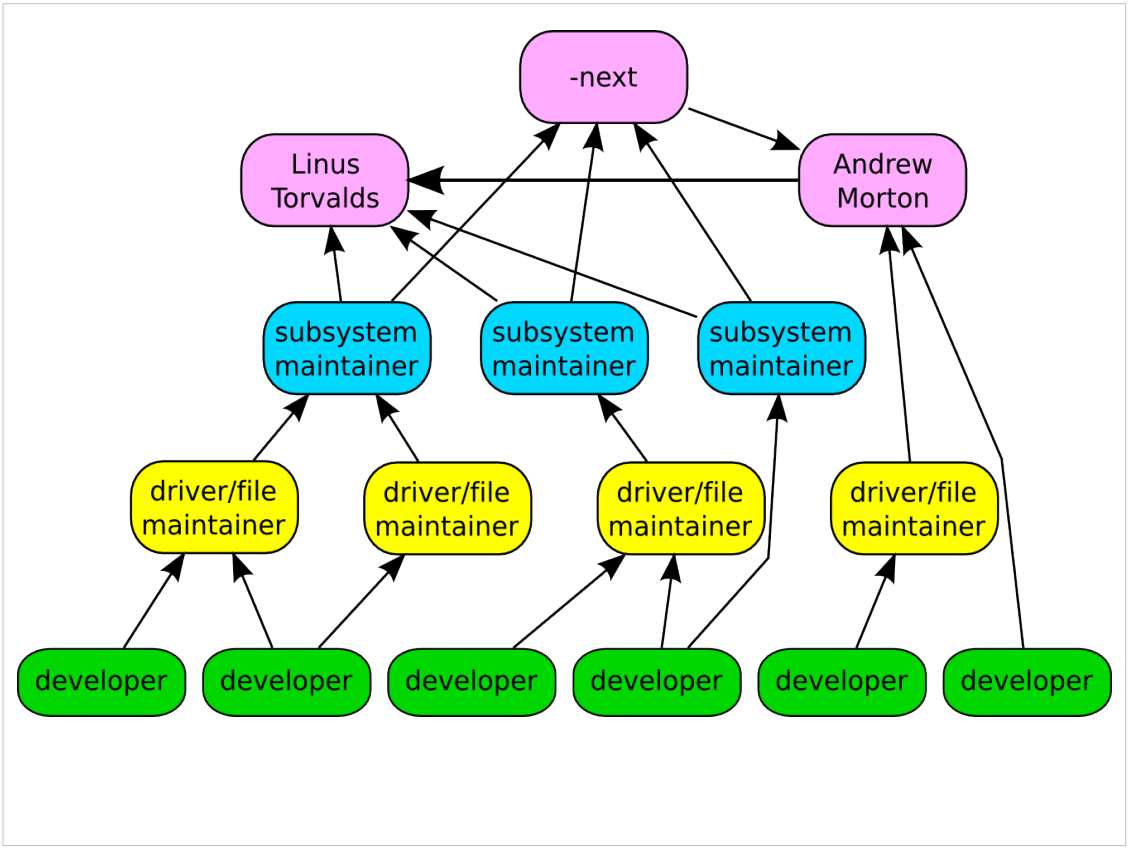
We have around 150 subsystem maintainers



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

We have about 150 different trees in the linux-next release.

Andrew Morton picks up patches that cross subsystems, or are missed by others, and releases his -mm kernels every few weeks. This includes the linux-next release at that time.

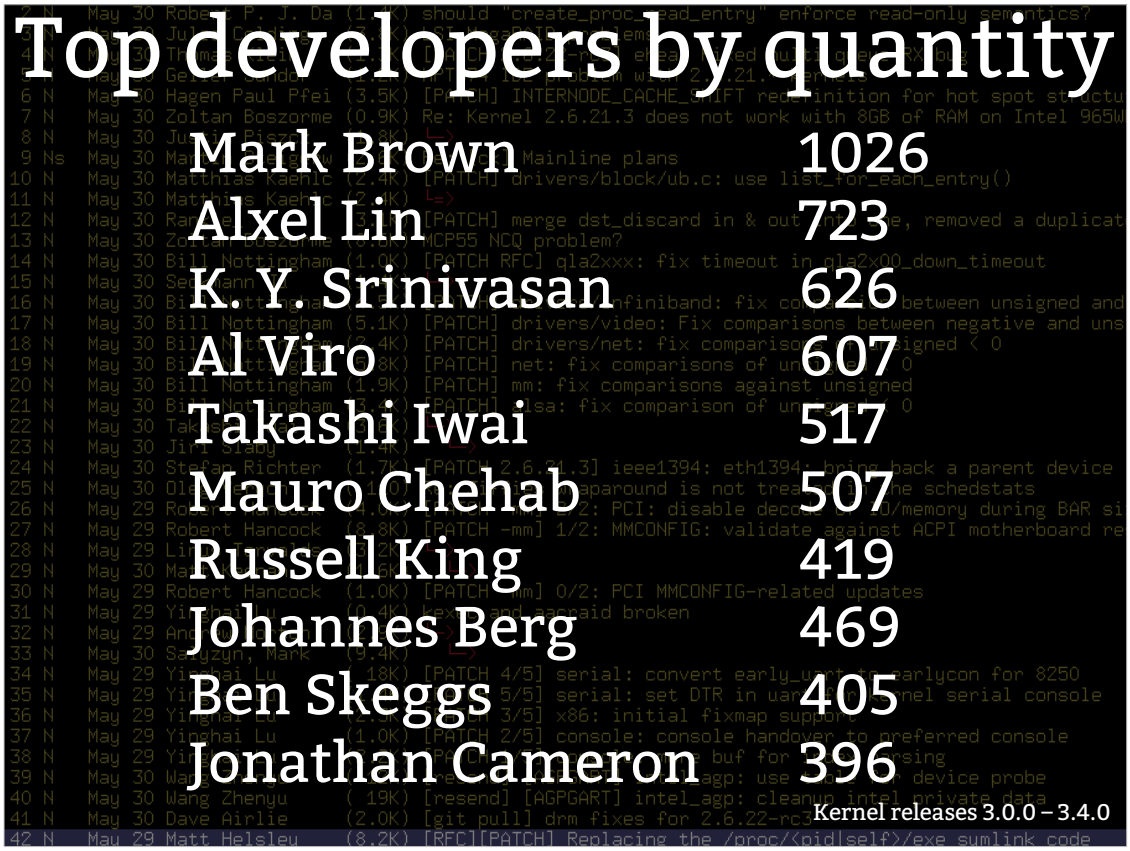


Every 3 months, when the merge window opens up, everything gets sent to Linus from the subsystem maintainers and Andrew Morton.

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, but that isn't always the case for various reasons.

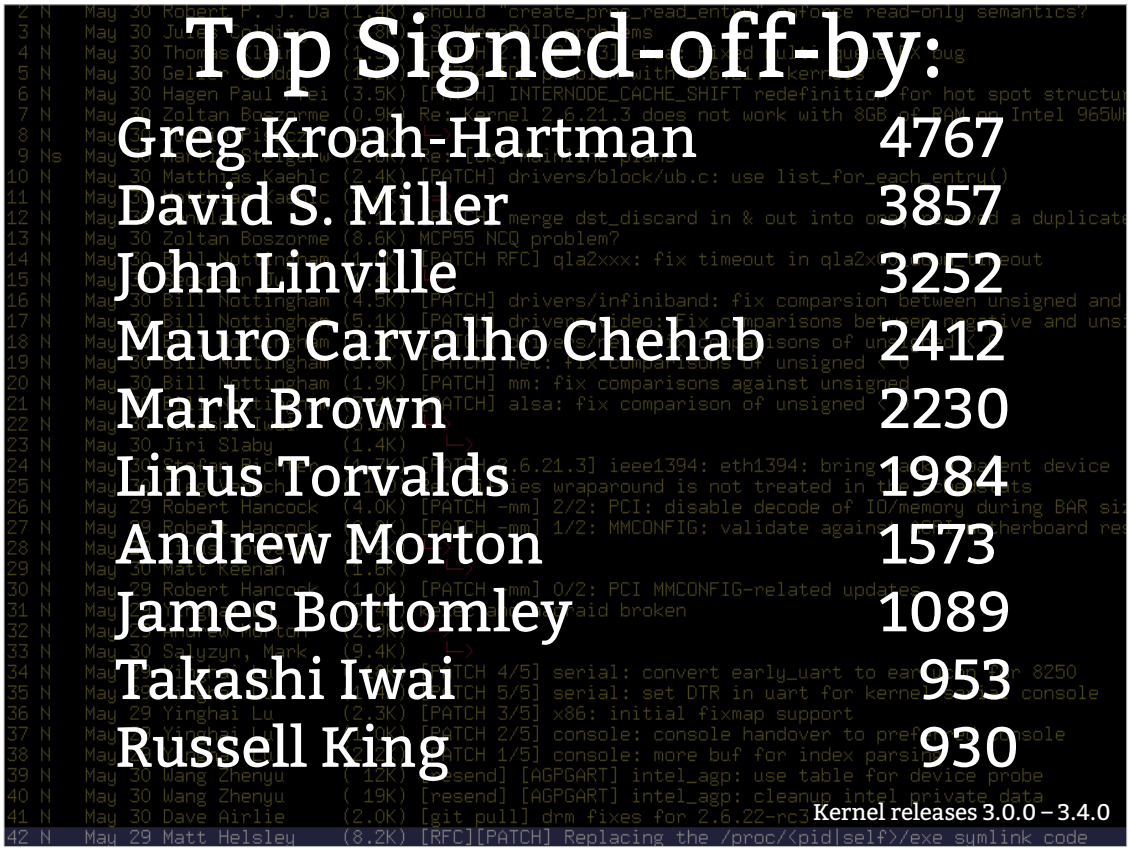
Linux-next can not just be sent to Linus as there are things in there that sometimes are not good enough to be merged just yet, it is up to the individual subsystem maintainer to decide what to merge.



Mark – embedded sound
Alexl – janitorial
KY – hyperv
Al – vfs and filesystem
Takashi – sound maintainer
Mauro – v4l
Russell – ARM maintainer
Johannes – wireless developer
Ben – nouveau developer
Jonathan – IIO

Top Signed-off-by:

Greg Kroah-Hartman	4767
David S. Miller	3857
John Linville	3252
Mauro Carvalho Chehab	2412
Mark Brown	2230
Linus Torvalds	1984
Andrew Morton	1573
James Bottomley	1089
Takashi Iwai	953
Russell King	930



Kernel releases 3.0.0 – 3.4.0

Greg – driver core, usb, staging
David – networking
John – wireless networking
Mauro - v4l
Linus – everything
Mark - embedded
Andrew – everything
James – SCSI
Takashi – sound
Russell - ARM

So you can view this as either 20% is done by non-affiliated people, or 80% is done by companies.

Now to be fair, if you show any skill in kernel development you are instantly hired.

Why this all matters: If your company relies on Linux, and it depends on the future of Linux 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 workloads and needs.

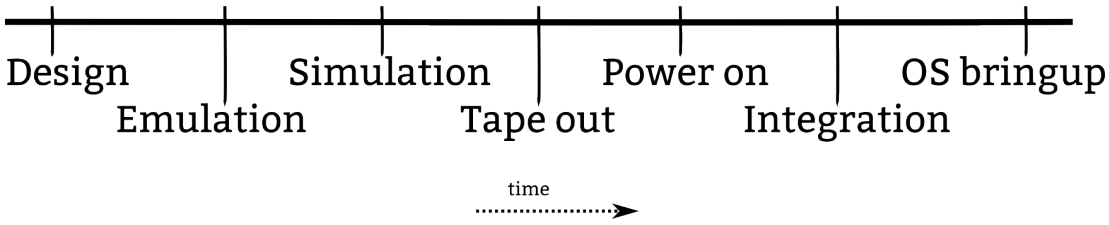
Samsung 980 patches
Qualcomm 707 patches

Who is funding this work?

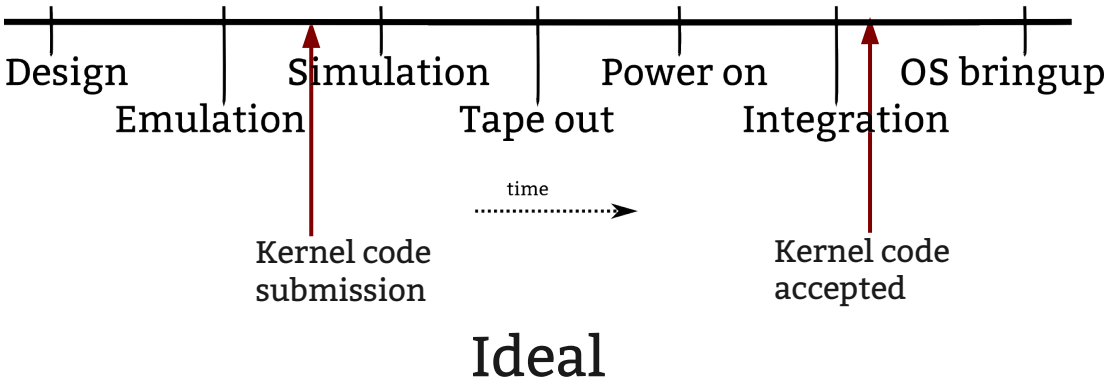
...	
23. Fujitsu	448
...	
43. Cisco	206
...	
59. HP	118
...	
230. Huawei	3

Kernel releases 3.0.0 – 3.4.0

Development Process



Development Process



Ideal

“Working upstream saves time and money”

Dan Frye – VP Open Systems, IBM
Dirk Hohndel – Chief Technologist, Intel

How to change

“Change or die”

How to change

Submit code early and often

No big code dumps.

They are hard to review, and even harder for you to modify and resend, slowing everything down and delaying any potential acceptance.

How to change

Send small pieces

How to change

Ask community for feedback

Some companies get Linux kernel community members together and discuss products and technologies directly with the senior engineers, no managers in the way.

Great feedback circle, the community gets to understand your products better, and no long explanation is needed when showing the code later, and the community gets to tell your engineers what they are doing wrong.

How to change

Act on it

Change products / roadmaps / features based on feedback. This makes Linux work better on your platforms which makes your platform better.

How to change

Remove legal hurdles

Let them contribute whatever they want and can.

Legal is to support the business, change the hurdles to be on the legal side, not the developers.

How to change

Force them to work in public

No internal mailing lists
All communication is done publically
Let them argue in public

How to change

Allow them to be the community

Your developers will become the maintainers, driving the future of Linux forward, always keeping your products in mind as things evolve and change.

Obligatory Penguin Picture