# Linux Kernel Development model

Greg Kroah-Hartman
gregkh@linux.com

37,085 files

14,770,000 lines

# 2,889 developers
# 358 companies

Kernel releases 2.6.36 – 3.1.0
October 2010 – October 2011

9,600 lines added
6,700 lines removed
2,100 lines modified

Kernel releases 2.6.36 – 3.1.0
October 2010 – October 2011

9,600 lines added
6,700 lines removed
2,100 lines modified

per day for all of 2011

Kernel releases 2.6.36 – 3.1.0
October 2010 – October 2011

# 5.44 changes per hour

Kernel releases 2.6.36 – 3.1.0
October 2010 – October 2011

# Kernel development changes

17 years ago...

# Linux 2.0.0

4 months later:

Linux 2.1.0

# Linux branches

Even number stable
Odd number development

# 848 days and 141 releases later...

# Linux 2.2.0

# 4 months later:

# Linux 2.3.0

# 604 days and 58 releases later...

# Linux 2.4.0

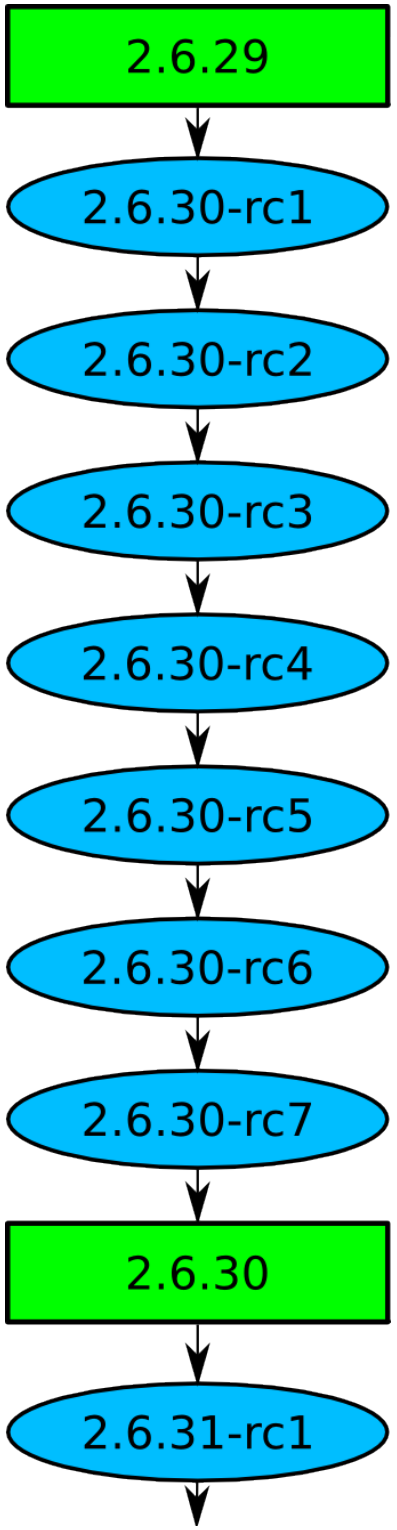
10 months later:

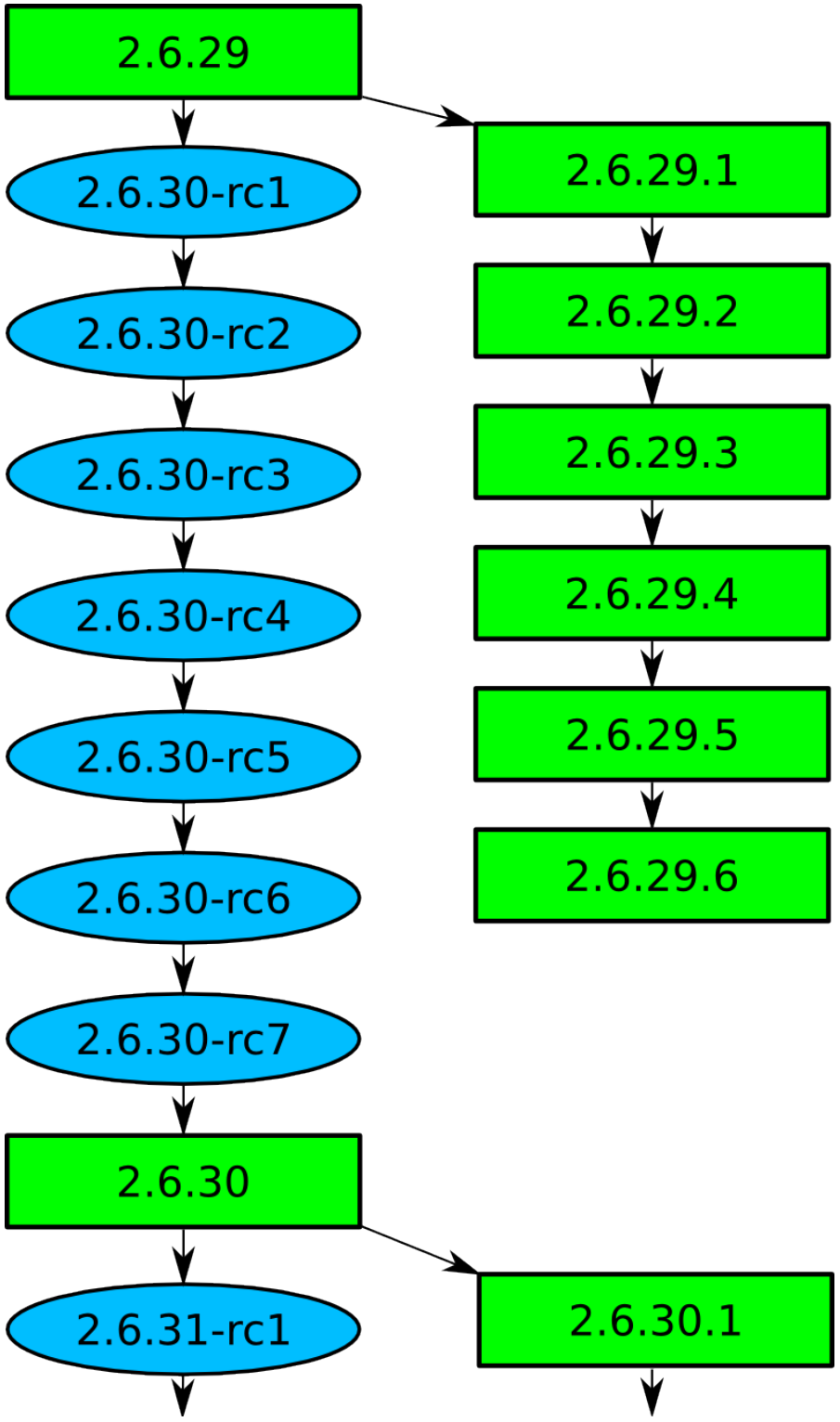
Linux 2.5.0

# 1057 days and 86 releases later...

# Linux 2.6.0


New release every
2¾ months
Kernel releases 2.6.20 – 3.1.0


2.6.29
2.6.30-rc1
2.6.30-rc2
2.6.30-rc3
2.6.30-rc4
2.6.30-rc5
2.6.30-rc6
2.6.30-rc7
2.6.30
2.6.31-rc1


2.6.29
2.6.30-rc1
2.6.30-rc2
2.6.30-rc3
2.6.30-rc4
2.6.30-rc5
2.6.30-rc6
2.6.30-rc7
2.6.30
2.6.31-rc1
2.6.29.1
2.6.29.2
2.6.29.3
2.6.29.4
2.6.29.5
2.6.29.6
2.6.30.1

# 2796 days and 39 releases later...

# Linux 3.0.0


developer
developer
developer
developer
developer

```
commit ecf85e481a716cfe07406439fdc7ba9526bbfaeb
Author:     Robert Jarzmik <robert.jarzmik@free.fr>
AuthorDate: Tue Apr 21 20:33:10 2009 -0700
Commit:     Greg Kroah-Hartman <gregkh@suse.de>
CommitDate: Thu Apr 23 14:15:31 2009 -0700

    USB: otg: Fix bug on remove path without transceiver

    In the case where a gadget driver is removed while no
    transceiver was found at probe time, a bug in
    otg_put_transceiver() will trigger.

    Signed-off-by: Robert Jarzmik <robert.jarzmik@free.fr>
    Acked-by: David Brownell <dbrownell@users.sourceforge.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@suse.de>

--- a/drivers/usb/otg/otg.c
+++ b/drivers/usb/otg/otg.c
@@ -43,7 +43,8 @@ EXPORT_SYMBOL(otg_get_transceiver);
 void otg_put_transceiver(struct otg_transceiver *x)
 {
-       put_device(x->dev);
+       if (x)
+               put_device(x->dev);
 }
```

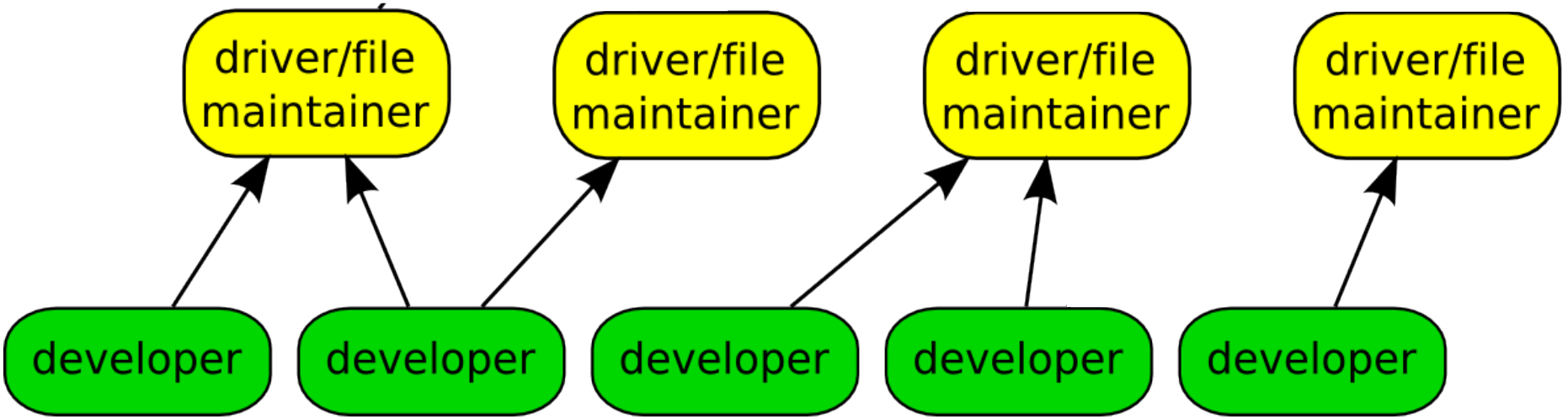
# Developer's Certificate of Origin

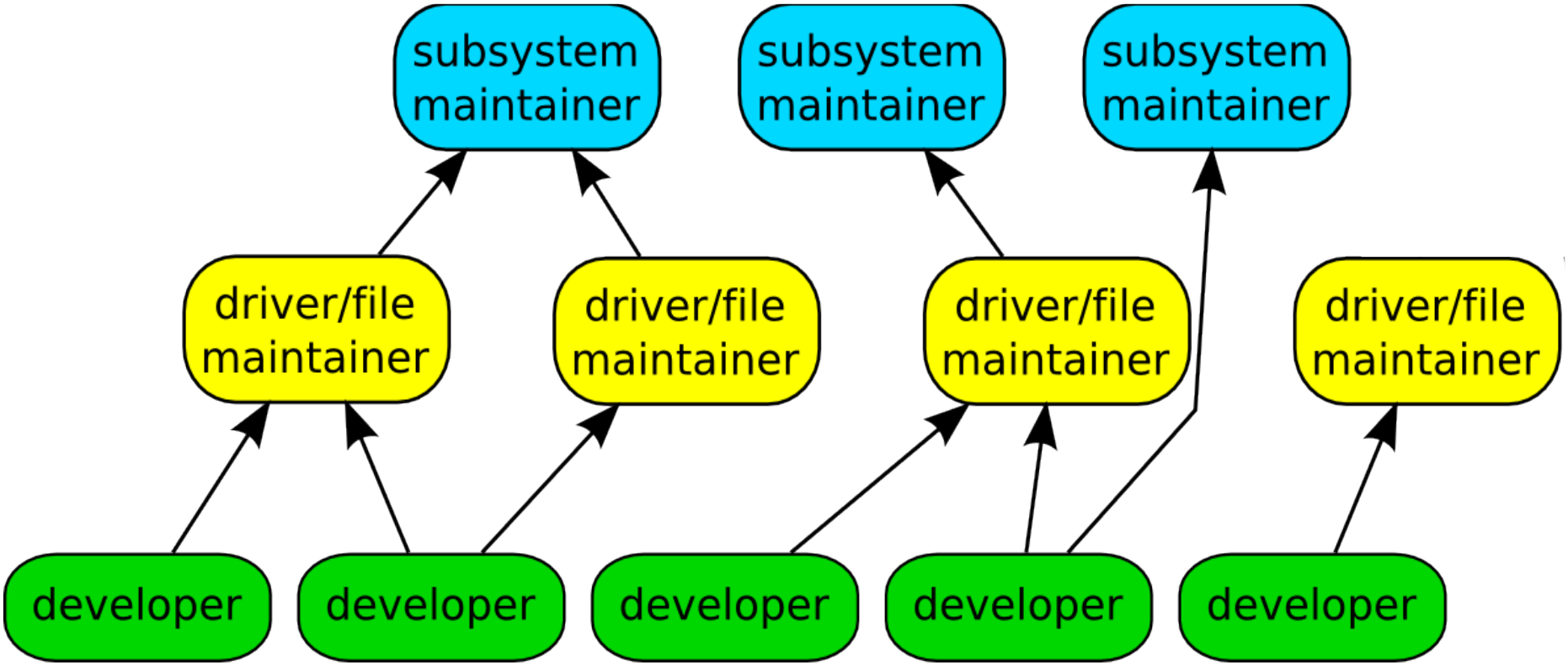
(a) I created this change; or

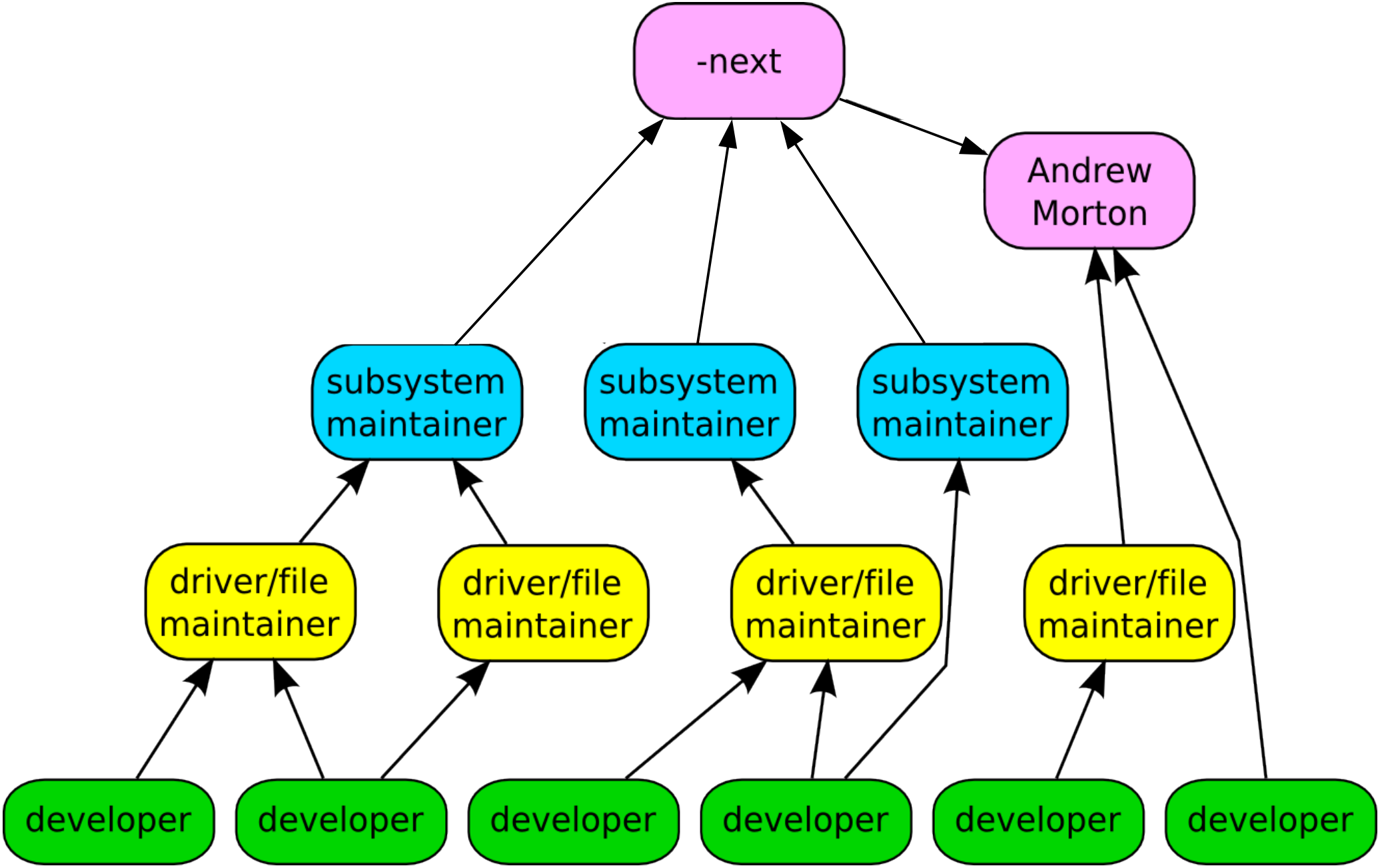
(b) Based this on a previous work with a compatible license; or

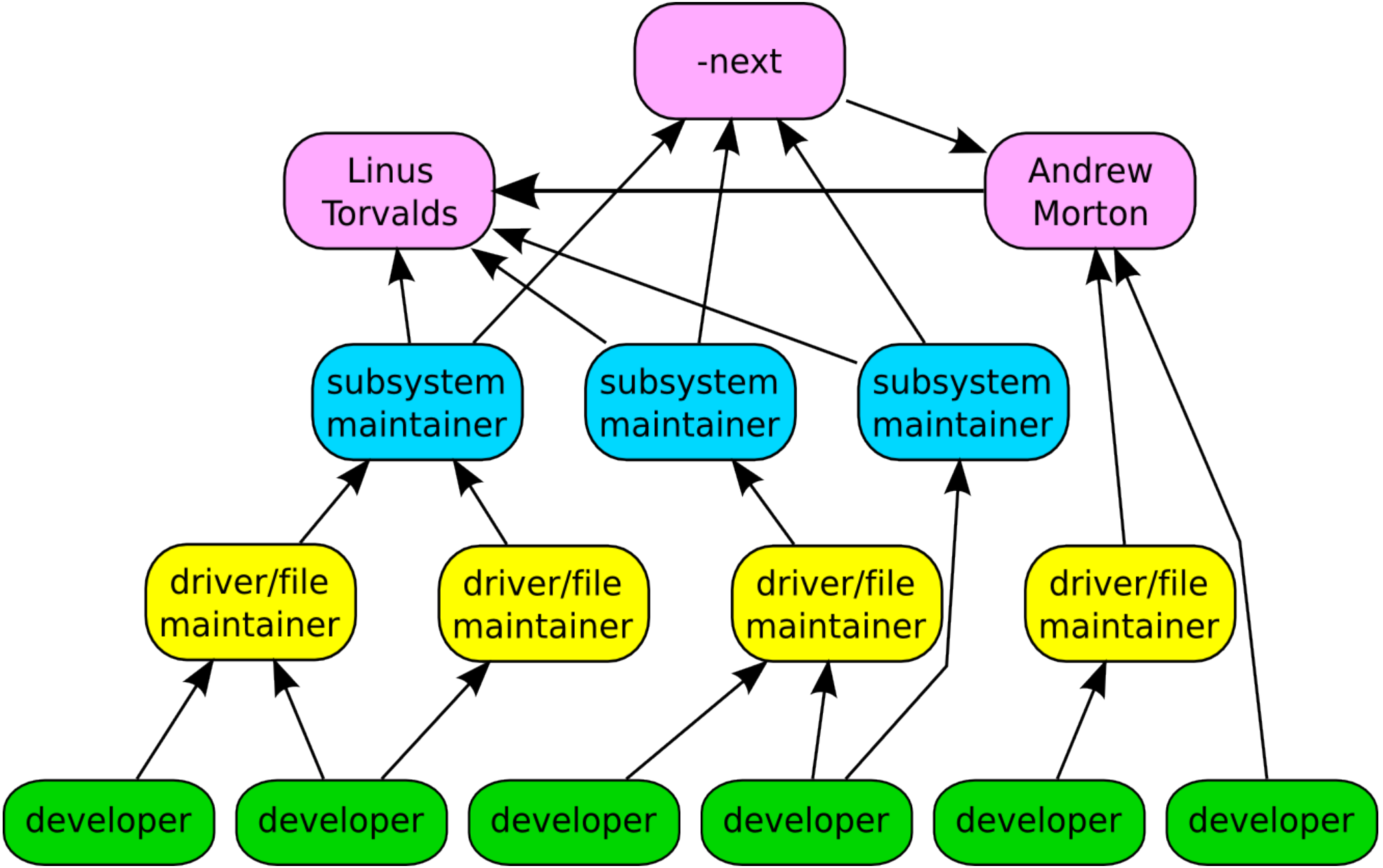
(c) Provided to me by (a), (b), or (c) and not modified

(d) This contribution is public.


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

| | |
|---|---|
| Chris Wilson | 566 |
| Joe Perches | 560 |
| Mark Brown | 532 |
| Eric Dumazet | 465 |
| Johannes Berg | 454 |
| David Miller | 407 |
| Takashi Iwai | 406 |
| Mauro Chehab | 401 |
| K. Y. Srinivasan | 401 |
| Al Viro | 396 |

Kernel releases 2.6.36 – 3.1.0

# Top Signed-off-by:

| | |
|---|---|
| Greg Kroah-Hartman | 4973 |
| David S. Miller | 4708 |
| John Linville | 2933 |
| Linus Torvalds | 2473 |
| Mauro Carvalho Chehab | 2195 |
| Andrew Morton | 2084 |
| Mark Brown | 1243 |
| James Bottomley | 1089 |
| Takashi Iwai | 955 |
| Russell King | 875 |

Kernel releases 2.6.36 – 3.1.0

# Who is funding this work?

| | |
|---|---|
| 1. | |
| 2. Red Hat | 11.4% |
| 3. Intel | 7.5% |
| 4. | |
| 5. Novell | 4.8% |
| 6. IBM | 3.9% |
| 7. Texas Instruments | 2.2% |
| 8. Broadcom | 1.9% |
| 9. Consultants | 2.3% |
| 10. Nokia | 1.8% |

Kernel releases 2.6.36 – 3.1.0

# Who is funding this work?

| | |
|---|---|
| 1. “Amateurs” | 15.5% |
| 2. Red Hat | 11.4% |
| 3. Intel | 7.5% |
| 4. Unknown Individuals | 6.3% |
| 5. Novell | 4.8% |
| 6. IBM | 3.9% |
| 7. Texas Instruments | 2.2% |
| 8. Broadcom | 1.9% |
| 9. Consultants | 2.3% |
| 10. Nokia | 1.8% |

Kernel releases 2.6.36 – 3.1.0

# Getting involved

# Getting involved

`Documentation/HOWTO`

`Documentation/ja_JP/HOWTO`

`Documentation/development-process`

# Getting involved

kernelnewbies.org

# Getting involved

## Linux Driver Project

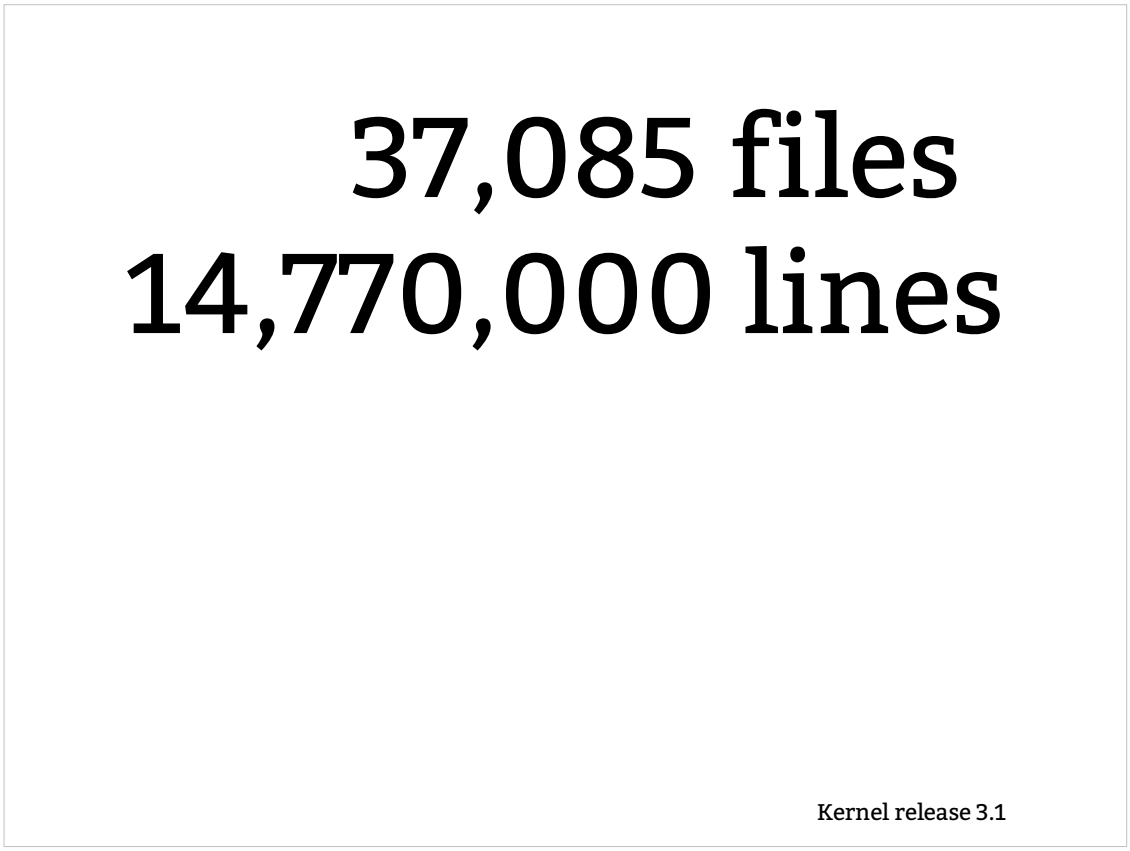
`drivers/staging/*/TODO`

ありがとう

# Linux Kernel Development model

Greg Kroah-Hartman
gregkh@linux.com

I'm going to discuss the history of how the Linux kernel was developed, how we handle the releases today, and how you can get involved in the process.




This was for the 3.1 kernel release, which happened October 24, 2011.

# 2,889 developers
# 358 companies

Kernel releases 2.6.36 – 3.1.0
October 2010 – October 2011

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.

9,600 lines added
6,700 lines removed
2,100 lines modified

Kernel releases 2.6.36 – 3.1.0
October 2010 – October 2011

9,600 lines added
6,700 lines removed
2,100 lines modified

per day for all of 2011

Kernel releases 2.6.36 – 3.1.0
October 2010 – October 2011

# 5.44 changes per hour

Kernel releases 2.6.36 – 3.1.0
October 2010 – October 2011

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.

# Kernel development changes

17 years ago...

# Linux 2.0.0

4 months later:

# Linux 2.1.0

# Linux branches

Even number stable
Odd number development

848 days and 141 releases later...

# Linux 2.2.0

4 months later:

# Linux 2.3.0

604 days and 58 releases later...

# Linux 2.4.0

10 months later:

Linux 2.5.0

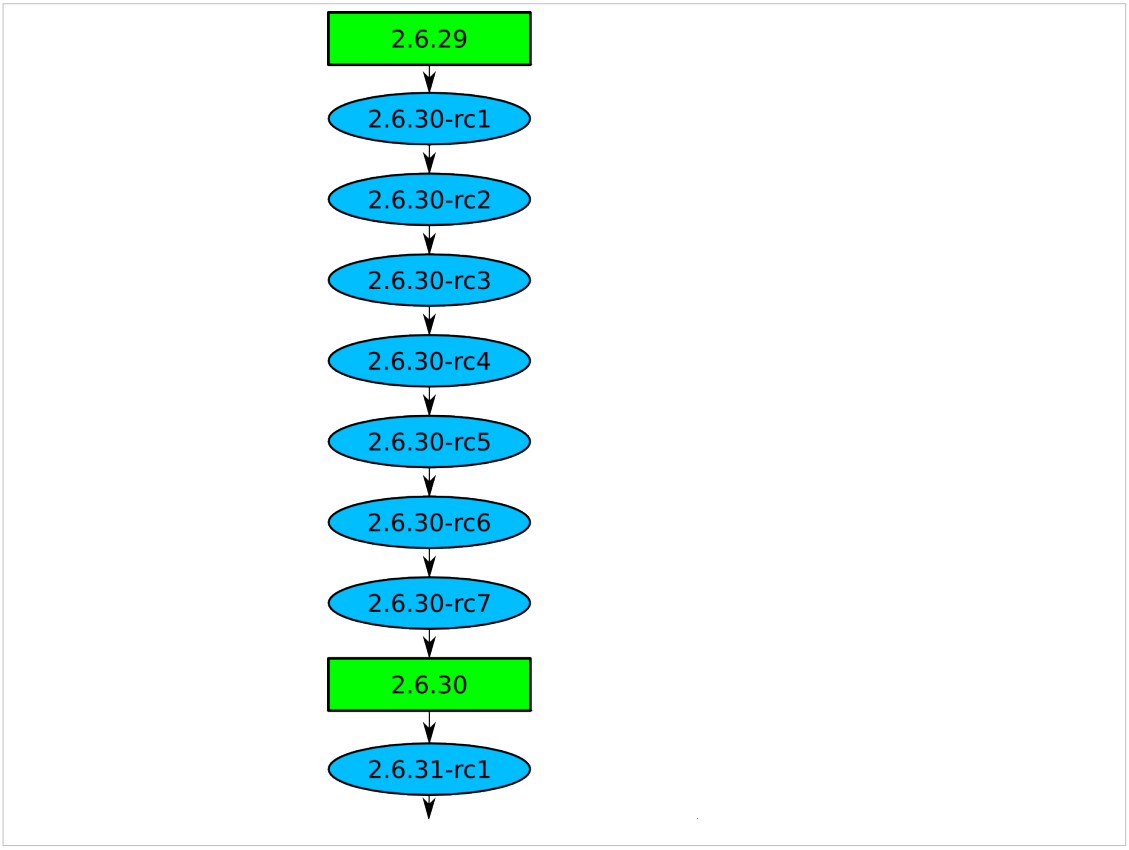
1057 days and 86 releases later...

# Linux 2.6.0

No more odd/even releases, every release is a “stable” one.

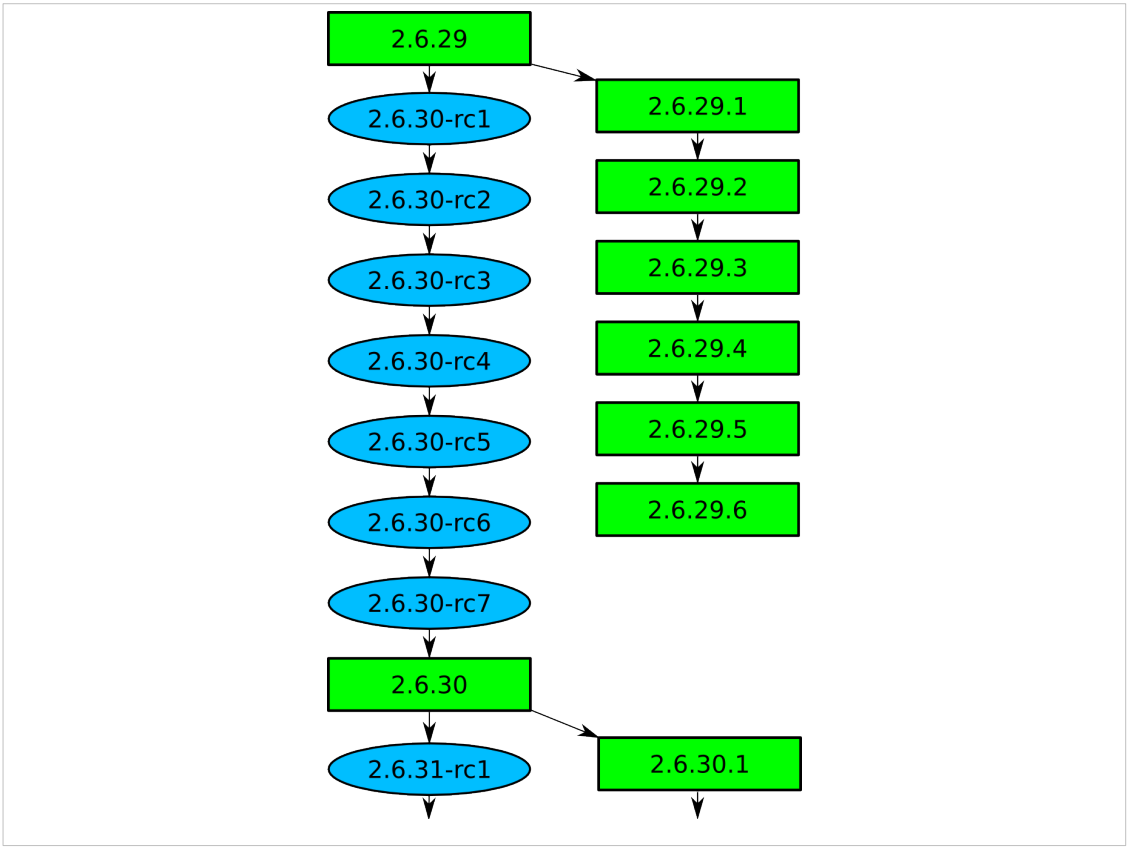
We don't want to live through the hell that the 2.5 development process was ever again.


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.

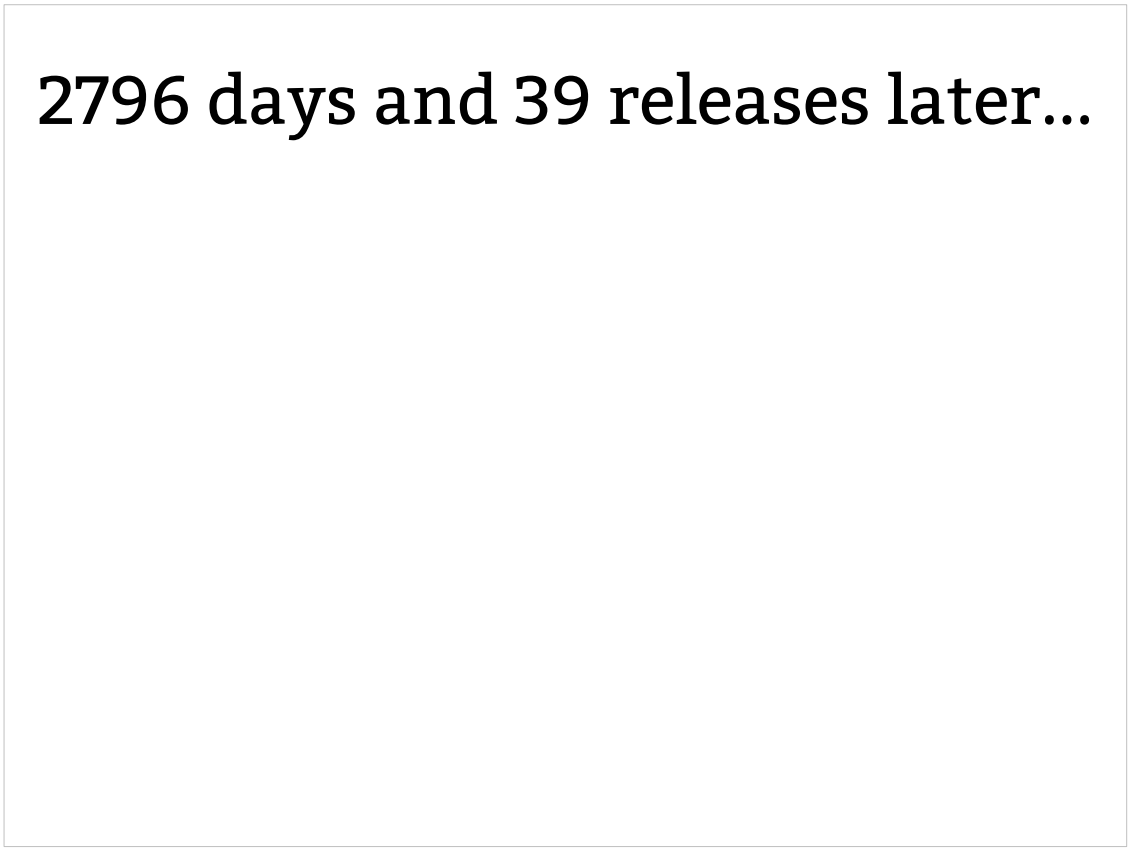



Greg takes the stable releases from Linus, and does stable releases with them, applying only fixes that are already in Linus's tree.

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, more about them later.



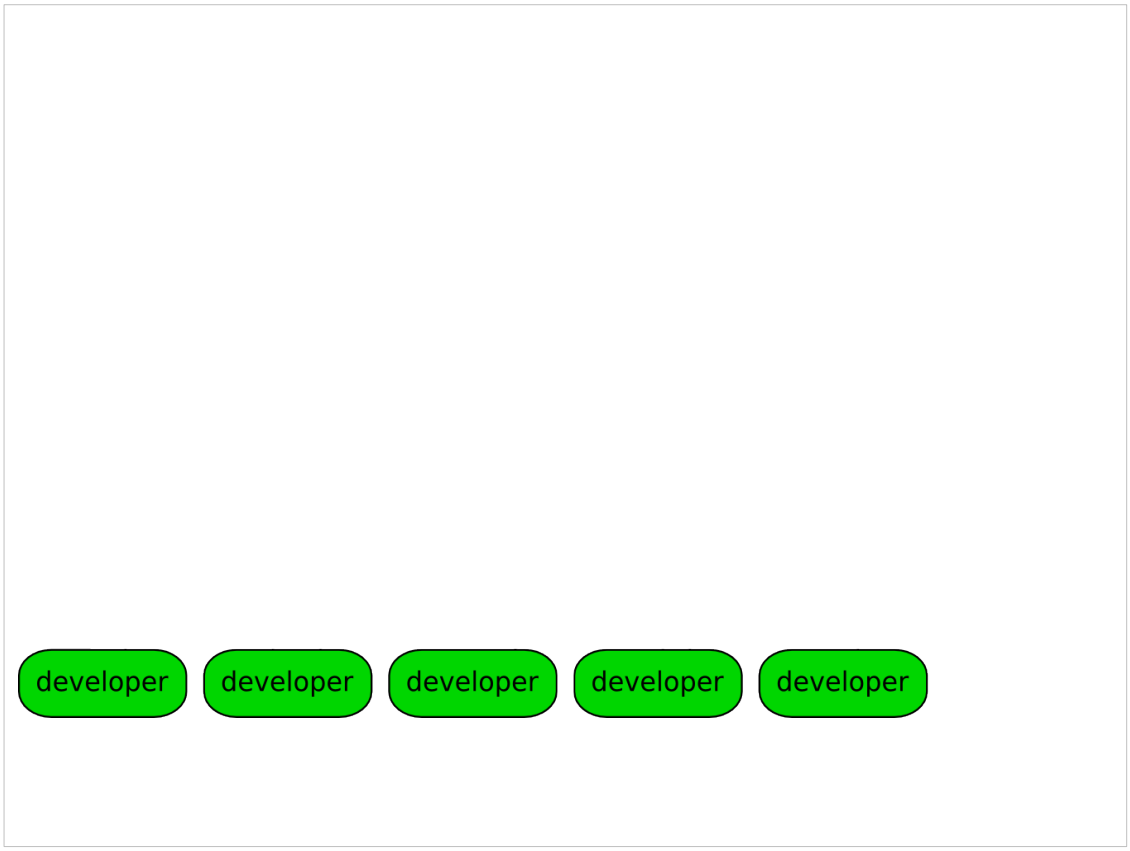

That's 7.6 years

# Linux 3.0.0

We got tired of the 2.6 prefix of the kernel releases.
39 releases was a lot, people started getting the numbers mixed up.

Nicely coincided with the 20th anniversary of Linux.

Release will be in the 3.x.y format.




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.

```
commit ecf85e481a716cfe07406439fdc7ba9526bbfaeb
Author:     Robert Jarzmik <robert.jarzmik@free.fr>
AuthorDate: Tue Apr 21 20:33:10 2009 -0700
Commit:     Greg Kroah-Hartman <gregkh@suse.de>
CommitDate: Thu Apr 23 14:15:31 2009 -0700

    USB: otg: Fix bug on remove path without transceiver

    In the case where a gadget driver is removed while no
    transceiver was found at probe time, a bug in
    otg_put_transceiver() will trigger.

    Signed-off-by: Robert Jarzmik <robert.jarzmik@free.fr>
    Acked-by: David Brownell <dbrownell@users.sourceforge.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@suse.de>

--- a/drivers/usb/otg/otg.c
+++ b/drivers/usb/otg/otg.c
@@ -43,7 +43,8 @@ EXPORT_SYMBOL(otg_get_transceiver);
 void otg_put_transceiver(struct otg_transceiver *x)
 {
-       put_device(x->dev);
+       if (x)
+               put_device(x->dev);
 }
```

This is an example of a patch.

It came from Robert, was acked by David, the maintainer at the time of the usb on-the-go subsystem, and then signed off by by me before it was commited to the kernel tree.

The change did one thing, it checked the value of the pointer before it was dereferenced, fixing a bug that would have crashed the kernel if it had been hit.

This is also a “blame” trail, showing who changed each line in the kernel, and who agreed with that change.

If a problem is found, these are the developers that you can ask about it.

Because of this, every line in the Linux kernel can be traced back to at least two developers who are responsible for it.

This is better than any other body of code.

## Developer's Certificate of Origin

(a) I created this change; or

(b) Based this on a previous work with a compatible license; or

(c) Provided to me by (a), (b), or (c) and not modified

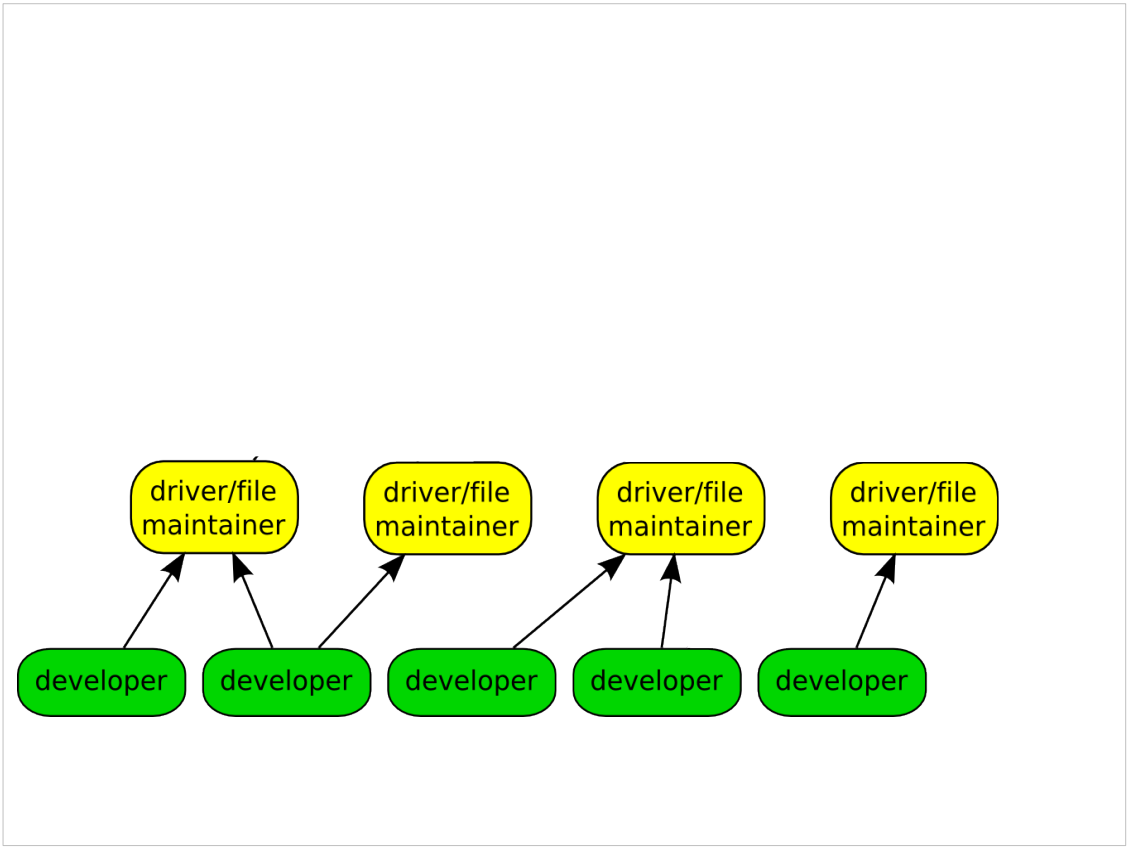
(d) This contribution is public.

This is what “Signed-off-by:” means.
All contributions to the Linux kernel have to agree to this, and every single patch has at least one signed-off-by line, usually all have at least two.

This is also a “blame” trail, showing who changed each line in the kernel, and who agreed with that change.

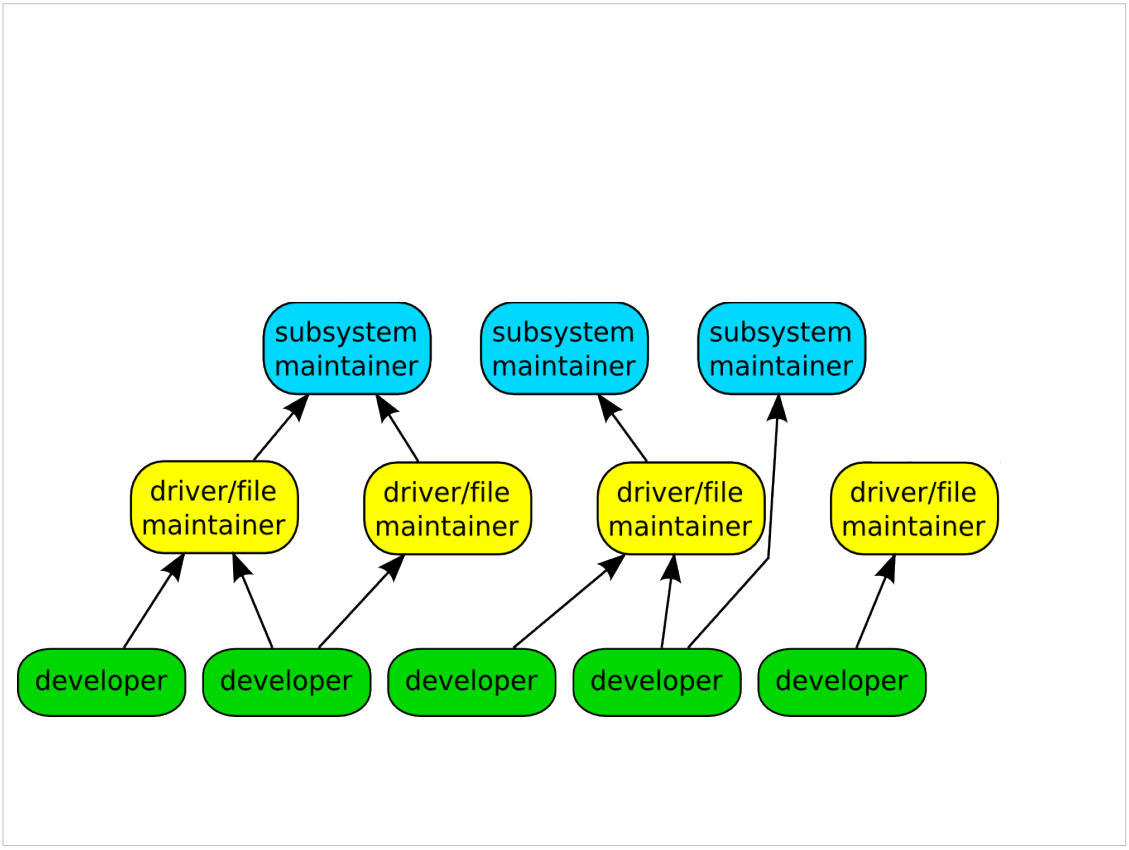
If a problem is found, this is the developers that you can ask about it.

Because of this, every line in the Linux kernel can be traced back to at least two developers who are responsible for it.
This is better than any other body of code.



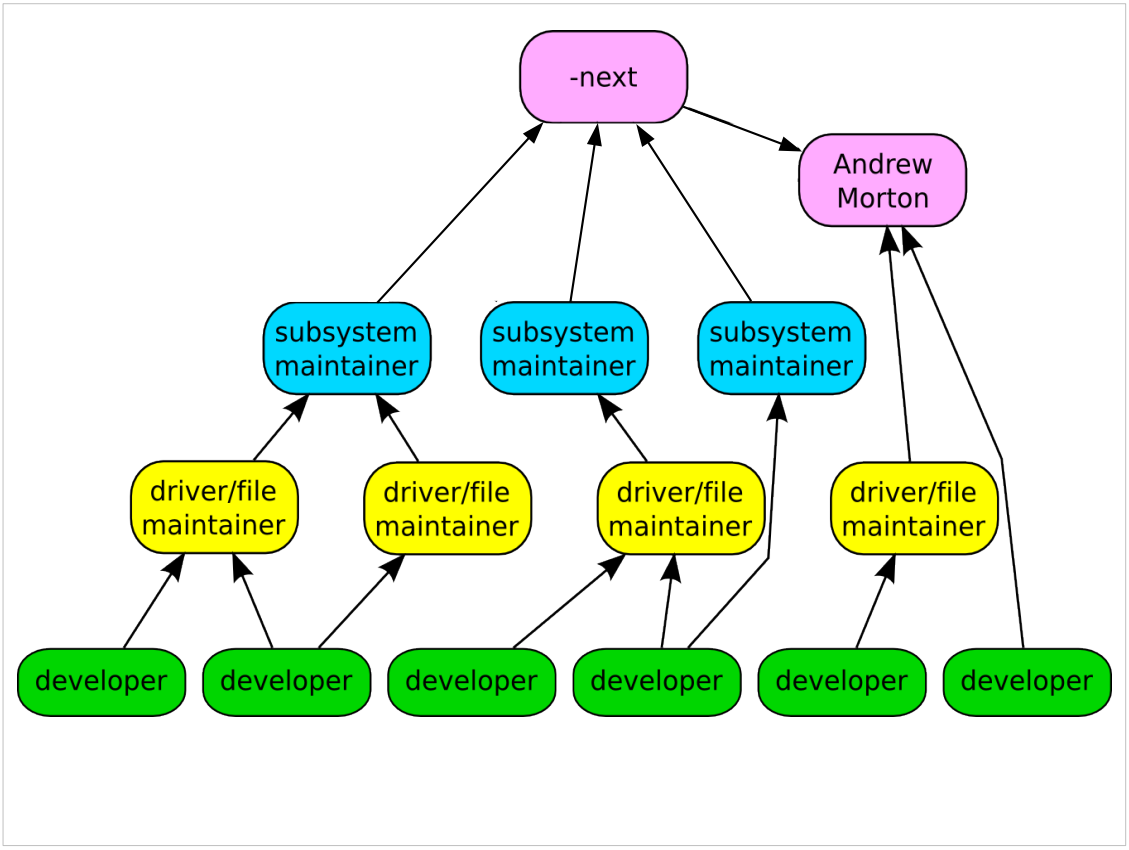

The developers send their patch to the maintainer of the file(s) that they have modified.

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.

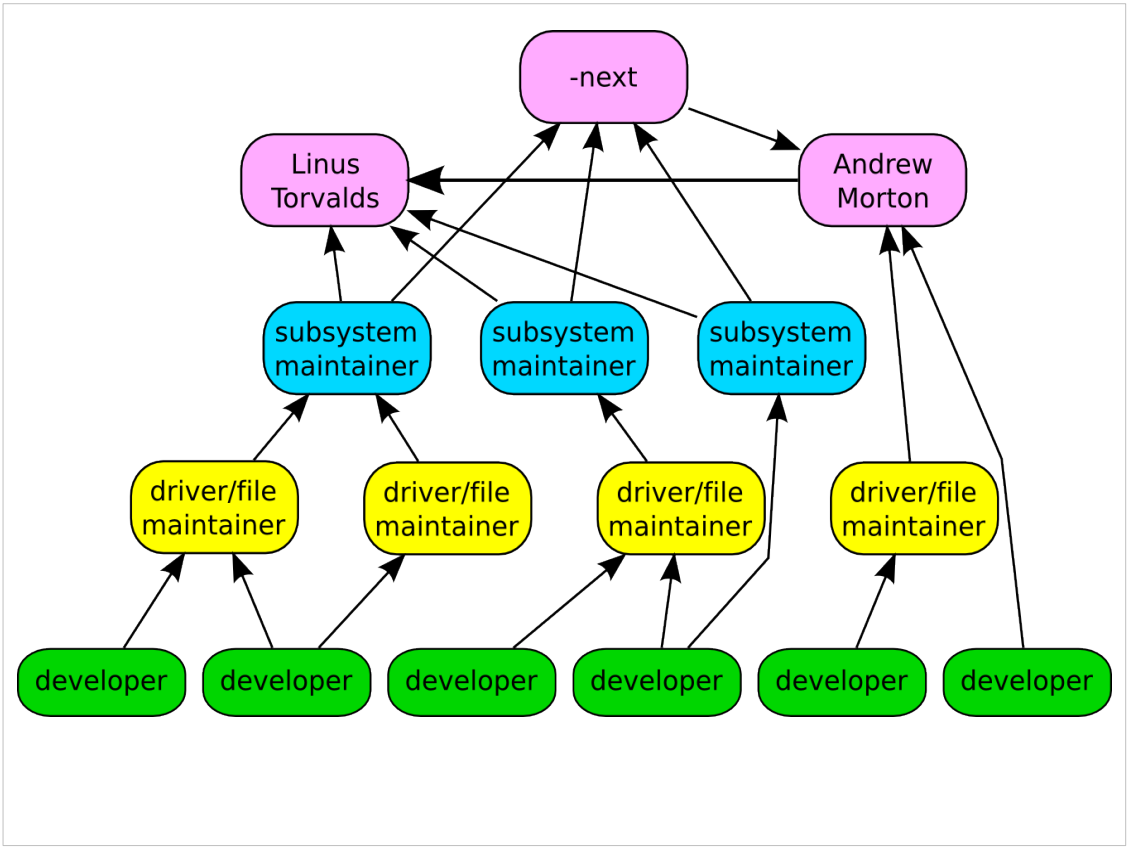
We have around 150 subsystem maintainers




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.

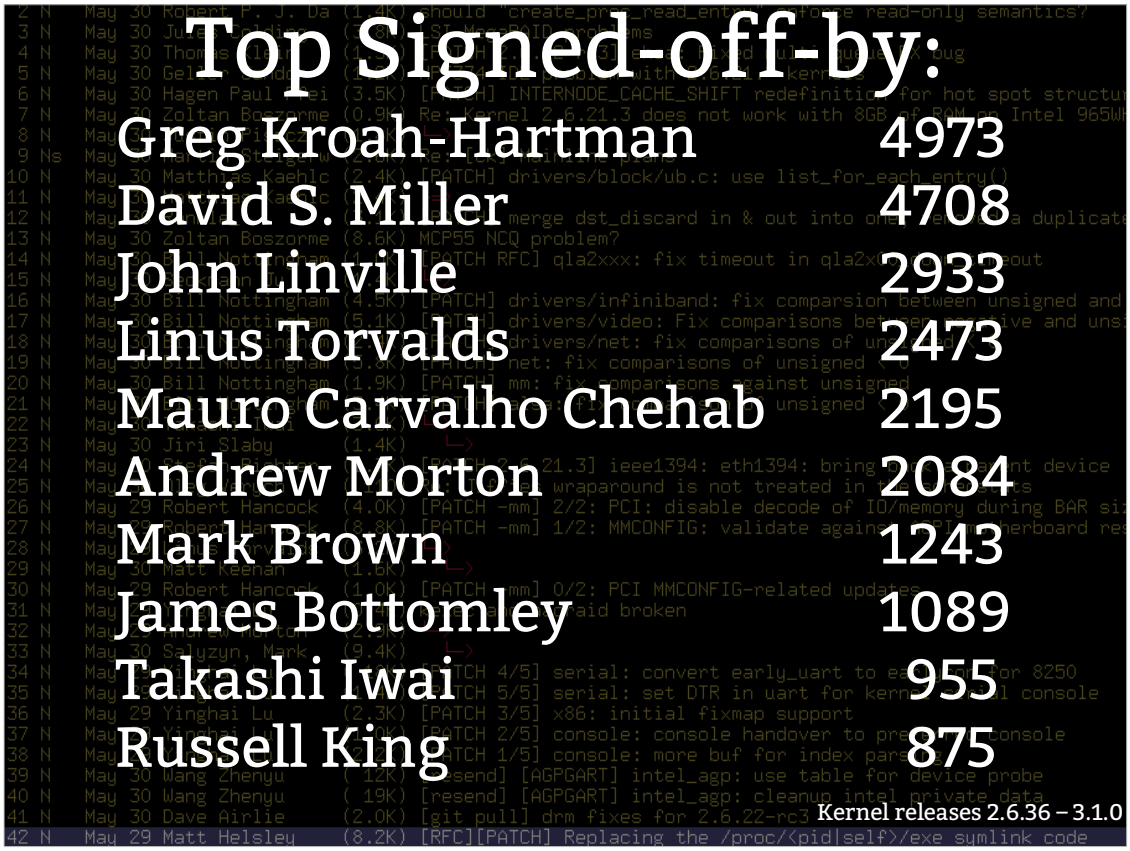
All of the patches merged to Linus should have been in the linux-next release, but that isn't always the case for various reasons.

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.


Chris Wilson – graphic drivers
Joe Perches – janitorial tasks
Mark Brown – embedded
Eric Dumazet – networking
Johannes Berg – wireless networking
David Miller – networking
Takashi Iwai – sound
Mauro Chehab – video for Linux
K.Y Srinivasan – hyperv drivers
Al Viro – vfs and filesystems

# Top Signed-off-by:

| | |
|---|---|
| Greg Kroah-Hartman | 4973 |
| David S. Miller | 4708 |
| John Linville | 2933 |
| Linus Torvalds | 2473 |
| Mauro Carvalho Chehab | 2195 |
| Andrew Morton | 2084 |
| Mark Brown | 1243 |
| James Bottomley | 1089 |
| Takashi Iwai | 955 |
| Russell King | 875 |



Kernel releases 2.6.36 – 3.1.0

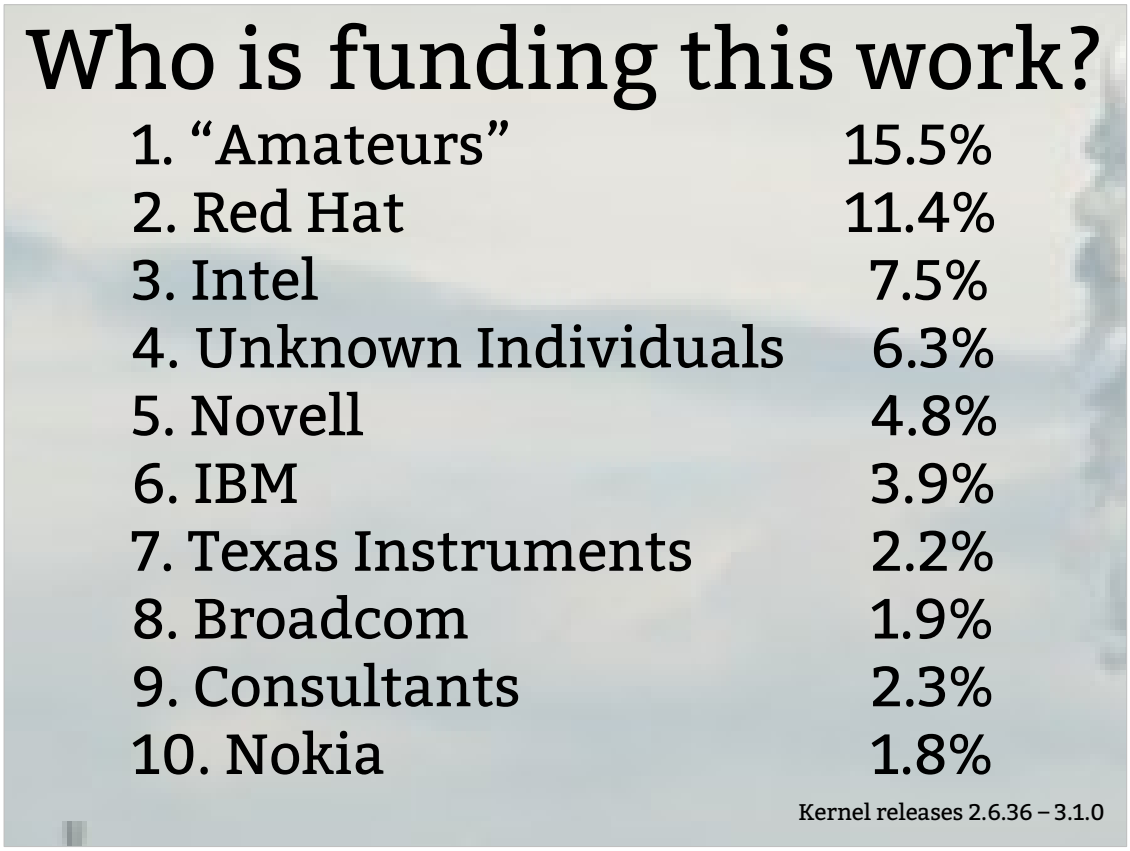
Greg – driver core, usb, staging
David – networking
John – wireless networking
Linus – everything
Mauro – v4l
Andrew – everything
Mark – embedded
James – SCSI
Takashi – sound
Russell - ARM

# Who is funding this work?

| | | |
|---|---|---|
| 1. | | |
| 2. | Red Hat | 11.4% |
| 3. | Intel | 7.5% |
| 4. | | |
| 5. | Novell | 4.8% |
| 6. | IBM | 3.9% |
| 7. | Texas Instruments | 2.2% |
| 8. | Broadcom | 1.9% |
| 9. | Consultants | 2.3% |
| 10. | Nokia | 1.8% |

Kernel releases 2.6.36 – 3.1.0

11 Oracle 799 (1.7%)
12 Samsung 796 (1.6%)
13 AMD 728 (1.5%)
14 Fujitsu 695 (1.4%)
15 Google 681 (1.4%)
16 Wolfson Microelectronics 609 (1.3%)
17 (Academia) 583 (1.2%)
18 Atheros Communications 572 (1.2%)
19 Pengutronix 523 (1.1%)
20 Analog Devices 500 (1.0%)
21 Microsoft 468 (1.0%)
22 Societe Francaise de Radiotelephone 465 (1.0%)
23 ST Ericsson 442 (0.9%)
24 Freescale 433 (0.9%)
25 QLogic 391 (0.8%)
26 MiTAC 363 (0.8%)
27 NTT 336 (0.7%)
28 Cisco 335 (0.7%)
29 Marvell 329 (0.7%)
30 Renesas Electronics 310 (0.6%)




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.

# Getting involved

# Getting involved

**Documentation/HOWTO**

**Documentation/ja_JP/HOWTO**

**Documentation/development-process**

These documents in the kernel source directory are the best place to start if you want to understand how the development process works, and how to get involved.

The HOWTO file has links to almost everything else you ever wanted.

More translations of the documentation files are always accepted, please send me and Shibata-san patches.

# Getting involved

kernelnewbies.org

http://www.kernelnewbies.org

# Getting involved

## Linux Driver Project

`drivers/staging/*/TODO`


ありがとう