Greg Kroah-Hartman
gregkh@linuxfoundation.org
github.com/gregkh/kernel-development
39,000 files
15,600,000 lines

Kernel release 3.5.0
2,841 developers
407 companies

Kernel releases 3.1.0 – 3.5.0
July 2011 – July 2012
10,600 lines added
8,100 lines removed
2,250 lines modified
10,600 lines added
8,100 lines removed
2,250 lines modified
every day

Kernel releases 3.1.0 – 3.5.0
July 2011 – July 2012
6.03 changes per hour

Kernel releases 3.1.0 – 3.5.0
July 2011 – July 2012
How we stay sane

Time based releases
Incremental changes
New release every 2\(\frac{3}{4}\) months
“Longterm kernels”

One picked per year
Maintained for two years

3.0 and 3.4
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.
Top developers by quantity

Mark Brown 1026
Axel Lin 879
Al Viro 681
Mauro Chehab 540
Russell King 496
Johannes Berg 486
Takashi Iwai 473
Dan Carpenter 410
Ben Skeggs 406
Greg Kroah-Hartman 404

Kernel releases 3.1.0 – 3.5.0
Top Signed-off-by:

Greg Kroah-Hartman  5474
David S. Miller  3986
John Linville  3123
Mauro Carvalho Chehab  2667
Mark Brown  2546
Linus Torvalds  2005
Andrew Morton  1632
James Bottomley  1027
David Airlie  987
Axel Lin  887

Kernel releases 3.1.0 – 3.5.0
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. Texas Instruments 3.6%
7. IBM 3.1%
8. Linaro 3.0%
9. Broadcom 2.6%
10. Consultants 2.3%

Kernel releases 3.1.0 – 3.5.0
Who is funding this work?

11. Wolfson Micro 2.2%
12. Google 2.0%
13. Samsung 2.0%
14. Oracle 1.7%
15. Ingics Technology 1.7%
16. Qualcomm 1.6%
17. Freescale 1.2%
18. Wind River 1.1%
19. Nokia 1.0%
20. Linux Foundation 0.9%

Kernel releases 3.1.0 – 3.5.0
Product Development

Design | Bringup | Integration | Testing | Release

(time)
Product Development

Design | Bringup | Integration | Testing | Release

Kernel code submission | Kernel code accepted

Ideal
“Working upstream saves time and money”

Dan Frye – VP Open Systems, IBM
Dirk Hohndel – Chief Technologist, Intel
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.
39,000 files
15,600,000 lines

This was for the 3.5 kernel release, which happened July 21, 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.

First one year timespan that we have surpassed 400 companies.
10,600 lines added
8,100 lines removed
2,250 lines modified

Kernel releases 3.1.0 – 3.5.0
July 2011 – July 2012
10,600 lines added
8,100 lines removed
2,250 lines modified
every day

Kernel releases 3.1.0 – 3.5.0
July 2011 – July 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.
How we stay sane

Time based releases
Incremental changes

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.
New release every $2\frac{3}{4}$ months

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...
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.

3.0 and 3.4 are the longterm kernel releases I am maintaining.

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.
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
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.
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.
<table>
<thead>
<tr>
<th>Developer</th>
<th>Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark Brown</td>
<td>1026</td>
</tr>
<tr>
<td>Axel Lin</td>
<td>879</td>
</tr>
<tr>
<td>Al Viro</td>
<td>681</td>
</tr>
<tr>
<td>Mauro Chehab</td>
<td>540</td>
</tr>
<tr>
<td>Russell King</td>
<td>496</td>
</tr>
<tr>
<td>Johannes Berg</td>
<td>486</td>
</tr>
<tr>
<td>Takashi Iwai</td>
<td>473</td>
</tr>
<tr>
<td>Dan Carpenter</td>
<td>410</td>
</tr>
<tr>
<td>Ben Skeggs</td>
<td>406</td>
</tr>
<tr>
<td>Greg Kroah-Hartman</td>
<td>404</td>
</tr>
</tbody>
</table>

Kernel releases 3.1.0 – 3.5.0

Mark – embedded sound
Axel – janitorial
Al – vfs and filesystem
Mauro – v4l
Russell – ARM
Johannes – intel wireless
Takashi sound
Dan – janitorial
Ben – nouveau
Greg – USB, staging, tty, etc.
Top Signed-off-by:

Greg Kroah-Hartman
David S. Miller
John Linville
Mauro Carvalho Chehab
Mark Brown
Linus Torvalds
Andrew Morton
James Bottomley
David Airlie
Axel Lin

Kernel releases 3.1.0 – 3.5.0

Greg – driver core, usb, staging
David – networking
John – wireless networking
Mauro - v4l
Mark – embedded sound
Linus - everything
Andrew – everything
James – SCSI
David - graphics
Axel - janitorial
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. Texas Instruments 3.6%
7. IBM 3.1%
8. Linaro 3.0%
9. Broadcom 2.6%
10. Consultants 2.3%

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.
Who is funding this work?

<table>
<thead>
<tr>
<th>Company</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wolfson Micro</td>
<td>2.2%</td>
</tr>
<tr>
<td>Google</td>
<td>2.0%</td>
</tr>
<tr>
<td>Samsung</td>
<td>2.0%</td>
</tr>
<tr>
<td>Oracle</td>
<td>1.7%</td>
</tr>
<tr>
<td>Ingics Technology</td>
<td>1.7%</td>
</tr>
<tr>
<td>Qualcomm</td>
<td>1.6%</td>
</tr>
<tr>
<td>Freescale</td>
<td>1.2%</td>
</tr>
<tr>
<td>Wind River</td>
<td>1.1%</td>
</tr>
<tr>
<td>Nokia</td>
<td>1.0%</td>
</tr>
<tr>
<td>Linux Foundation</td>
<td>0.9%</td>
</tr>
</tbody>
</table>

Kernel releases 3.1.0 – 3.5.0

- Samsung 1047 patches
- LF – 501 patches
- Qualcomm 707 patches
Product Development

Design | Bringup | Integration | Testing | Release

(time)
Product Development

Design  Bringup  Integration  Testing  Release

Kernel code submission  Kernel code accepted

Ideal
“Working upstream saves time and money”

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