Last year in the Linux Kernel

Greg Kroah-Hartman

47,000 files
18,900,000 lines

3,483 developers
439 companies

Kernel releases 3.12.0 – 3.17.0
November 2013 – October 2014

8,300 lines added
4,650 lines removed
1,900 lines modified

Kernel releases 3.12.0 – 3.17.0
November 2013 – October 2014

8,300 lines added
4,650 lines removed
1,900 lines modified

Every Day

Kernel releases 3.12.0 – 3.17.0
November 2013 – October 2014

7.8 changes per hour

Kernel releases 3.12.0 – 3.17.0
November 2013 – October 2014

9.5 changes per hour

3.16 release

Notable Changes

Notable Changes

btrfs offline
AMD Radeon boost
GPU switching
separate GPU device nodes
timerless multiasking
RAID5 multithreading
lockref
better OOM
XFS recursion
tty lock rework
IPC lock rework
seqlock
idr
inittmpfs
restricted sysfs
Cachefiles
soft module dependancies
aio ring page migration
aio deferred completion
fair zone allocator
hugepage node migration
ssd block allocation
swap per-cpu allocation
swap discard async
detect hybrid MBRs
dm cache block size limits
btrfs compressed extents
btrfs UUID rework
ext4 pre-cacheing
ext4 external journal
ext4 corrupt marking
ext3 external journal
xfs object readahead
f2fs inline xattrs
f2fs garbage control
pstore compression
pstore decompression
pstore extensions
ceph punch hole
hfs ACLS
isofs RW rework
udf RW rework
TCP_NOTSENT_LOWAT
TSO autosizing
tcp_syncookies
tcp throughput increase
TS-ECR for RTT
use RTT for RTO
ipv6 UDP tunnel segment
ipv6 RFC 6980 & 3810
bridge multicast snoop
macvland fdb
physical port sysfs
igmp unsolicited report
tcp_probe ipv6
netfilter ipv6 SYNPROXY
reduced txpower 5/10Mhz
5/10Mhz scanning IBSS
openvswitch SCTP
pkt_sched fair queueing
usbnet USB3 throughput
OMAP SHAM
OMAP SHA384/SHA512
NEON XOR
vfio-pci hot reset
64bit PV guest NMIs

3.12 release

Faster low-level locks

3.12 release

Timerless multitasking

3.13 release

Multi-queue block layer

3.13 release

nftables

3.13 release

nfc payment support

3.14 release

Deadline scheduler

3.14 release

Kernel address space rand

3.14 release

Antibufferbloat packet sc

3.15 release

Faster resume

3.15 release

Faster erase and zero portions of a file

3.15 release

File cross-rename

3.15 release

FUSE write improvements

3.16 release

Unified control group heirachy

3.16 release

TCP fast open on IPV6

3.16 release

CONFIG_USB_DEBUG rem

3.16 release

32bit VDSO on 64bit

3.17 release

File sealing - memfd

3.17 release

Thunderbolt on Apple hardware

3.17 release

getrandom()

3.17 release

Signed kexec kernels

3.18 release

unionfs

3.18 release

bfp() syscall

3.18 release

Small network packet imp

future release?

Live kernel patching

Sometime in 2014

future release?

kdbus

Sometime in 2014

future release?

kselftests

Sometime in 2014

future release?

O_BENEATH

Sometime in 2014

future release?

cgroup namespaces

Sometime in 2014

future releases?

bfp() syscall

Sometime in 2014

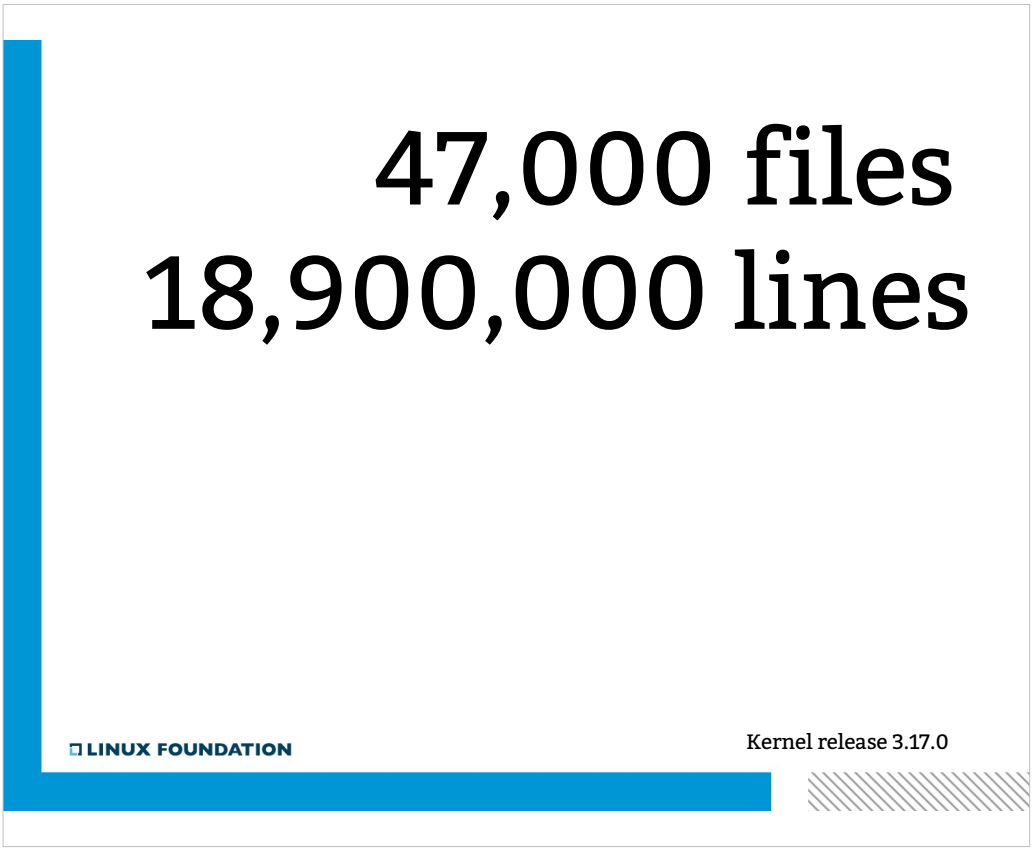
Who is funding this work?

1. Intel	10.6%
2. “Amateurs”	10.3%
3. Red Hat	8.4%
4. Unknown Individuals	7.3%
5. Linaro	5.6%
6. Samsung	4.4%
7. IBM	3.0%
8. SuSE	3.0%
9. Consultants	2.6%
10. Texas Instruments	2.4%

Who is funding this work?

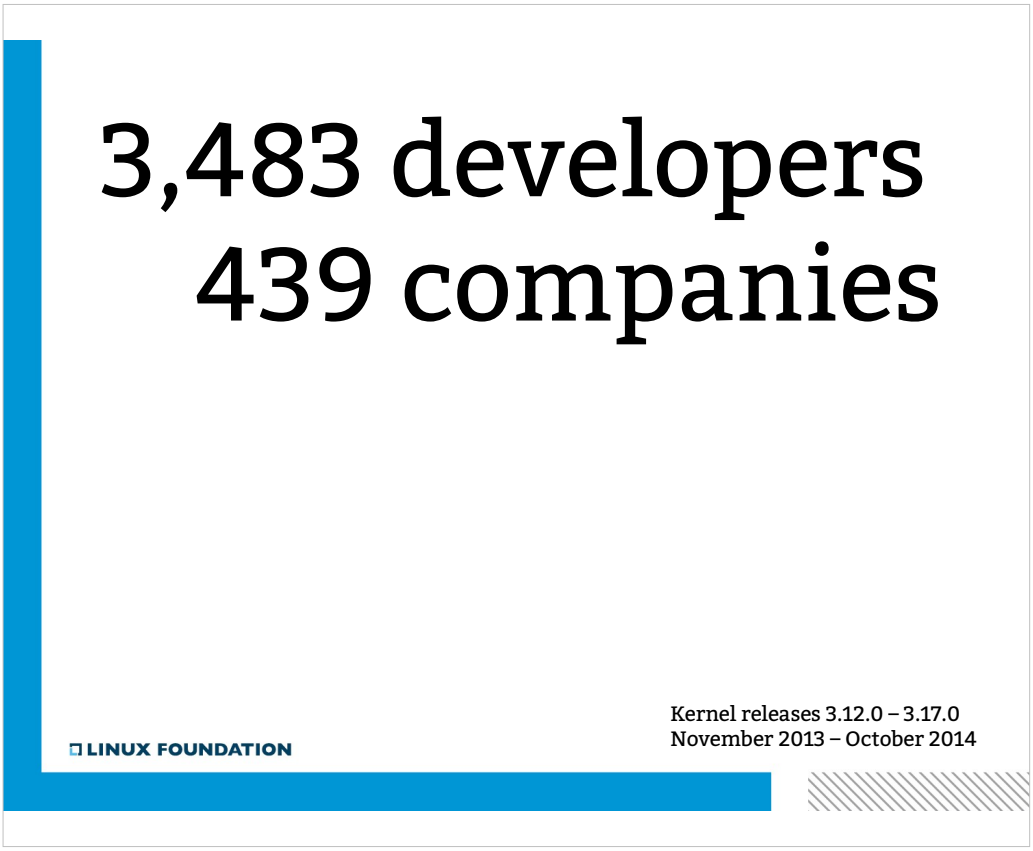
11. Vision Engraving	2.0%
12. Google	2.0%
13. Renesas	2.0%
14. Freescale	1.8%
15. Free Electrons	1.6%
16. Nvidia	1.2%
17. FOSS OPFW	1.2%
18. Oracle	1.2%
19. AMD	1.0%
20. Huawei	0.9%

I'm going to discuss what the Linux kernel community did over the past year, provide some example features, and some guesses as to what will be done in the future.



This is for the 3.17 kernel release, which happened on October 5, 2014.

This kernel is one of only 2 kernel releases that went down in size. Usually we grow at the constant rate of 1.5%, but this release shrunk due to 200 thousand lines being removed by one of the OPW interns, deleting a number of unused and unneeded device drivers.

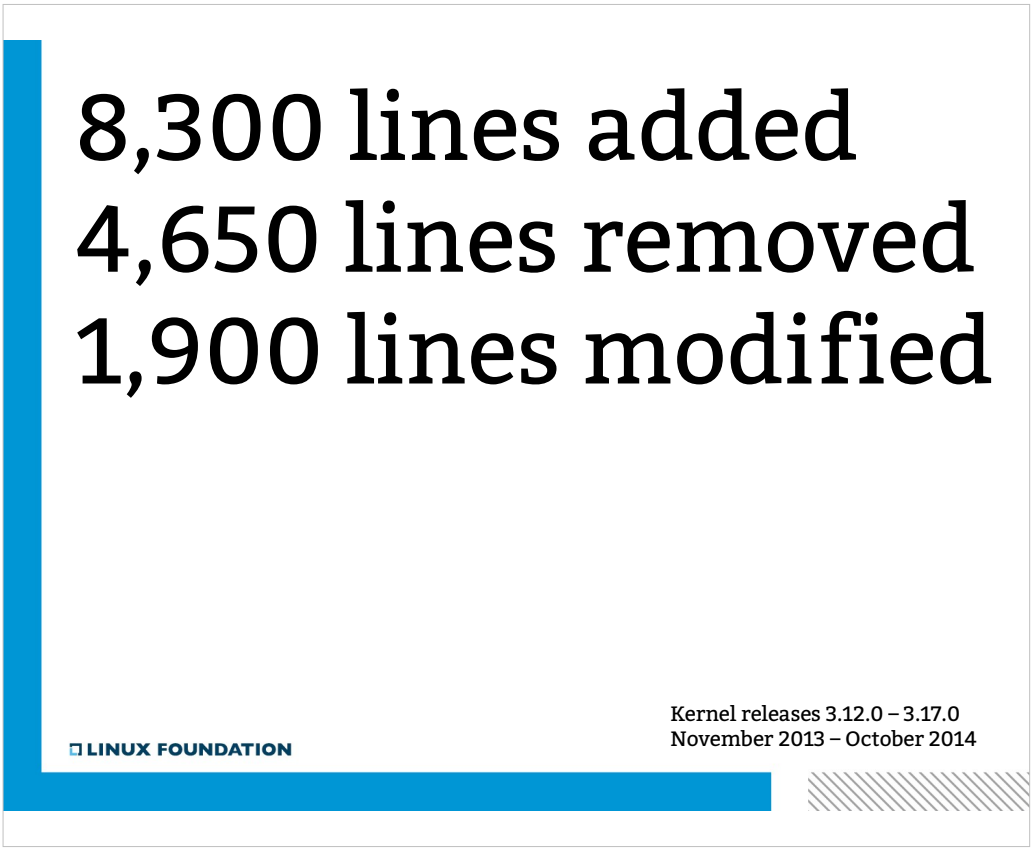


This is for the past year's worth of development, from the 3.12 through the 3.17 kernel release.

This makes the Linux kernel the largest software project ever.

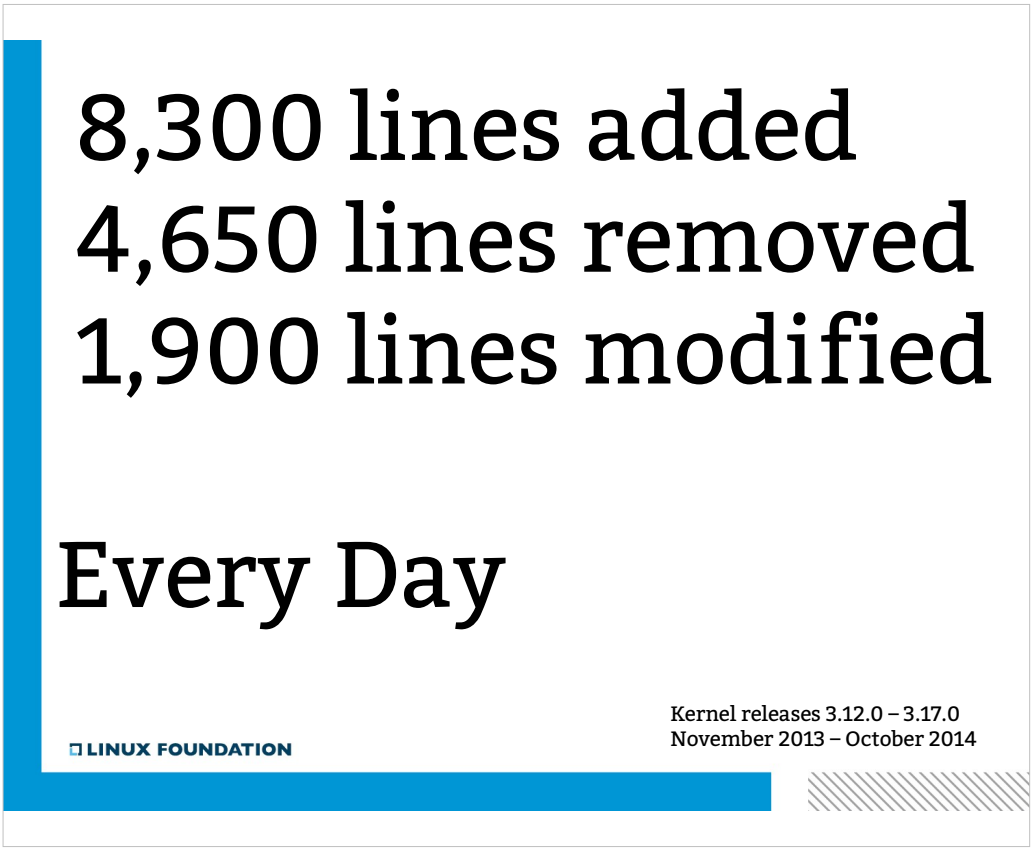
This is just the number of companies that we know about, there are more that have contributed, I have not kept up to date with tracking the number of companies.

We have surpassed over 400 different companies for the past 2 years. These numbers keep getting larger.



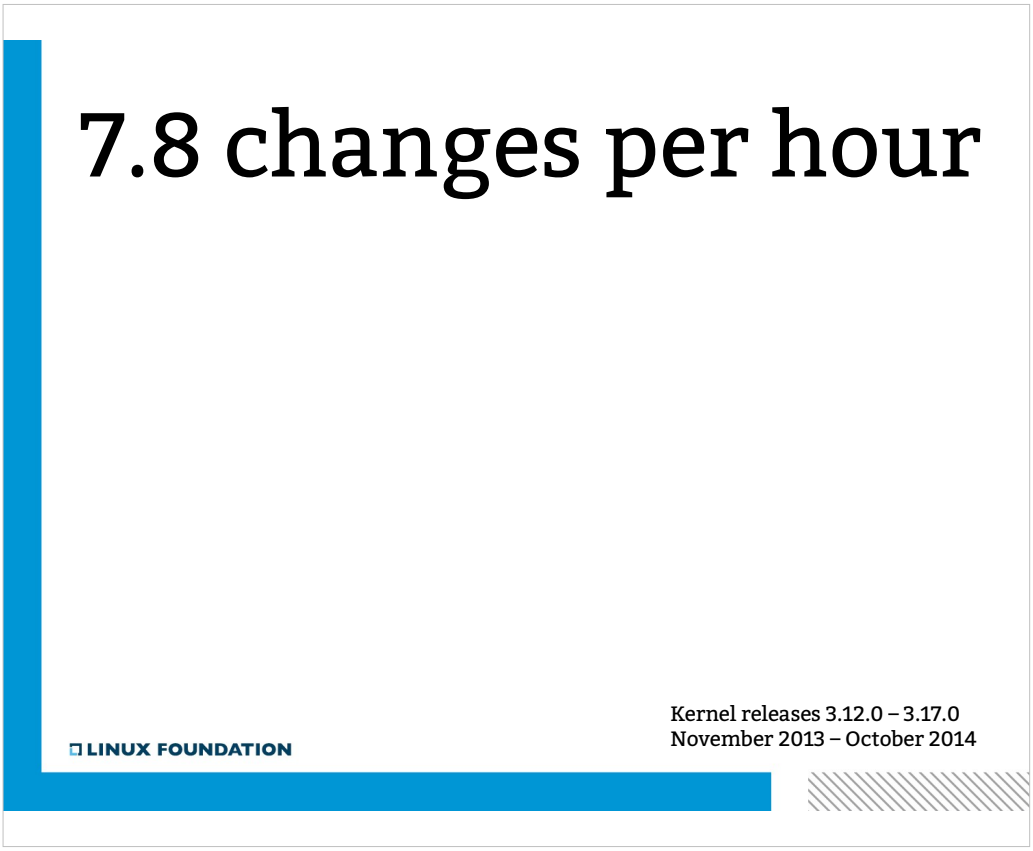
This is our current rate of change, which doesn't seem all that bad.

Untill...



You relealize the unite of change.

This happens every day, and it keeps going up.



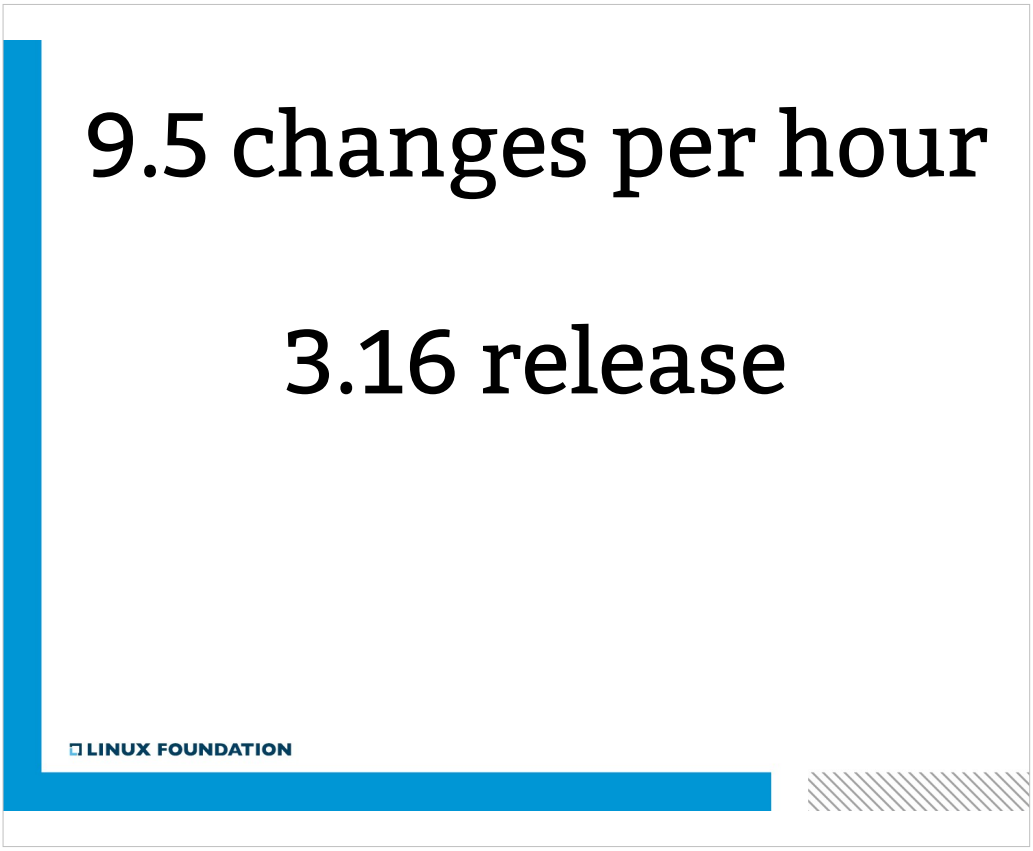
This is 24 hours a day, 7 days a week, for the past year.

We have increased this rate every year, an amazing rate of change.

All of these changes are for the whole kernel.

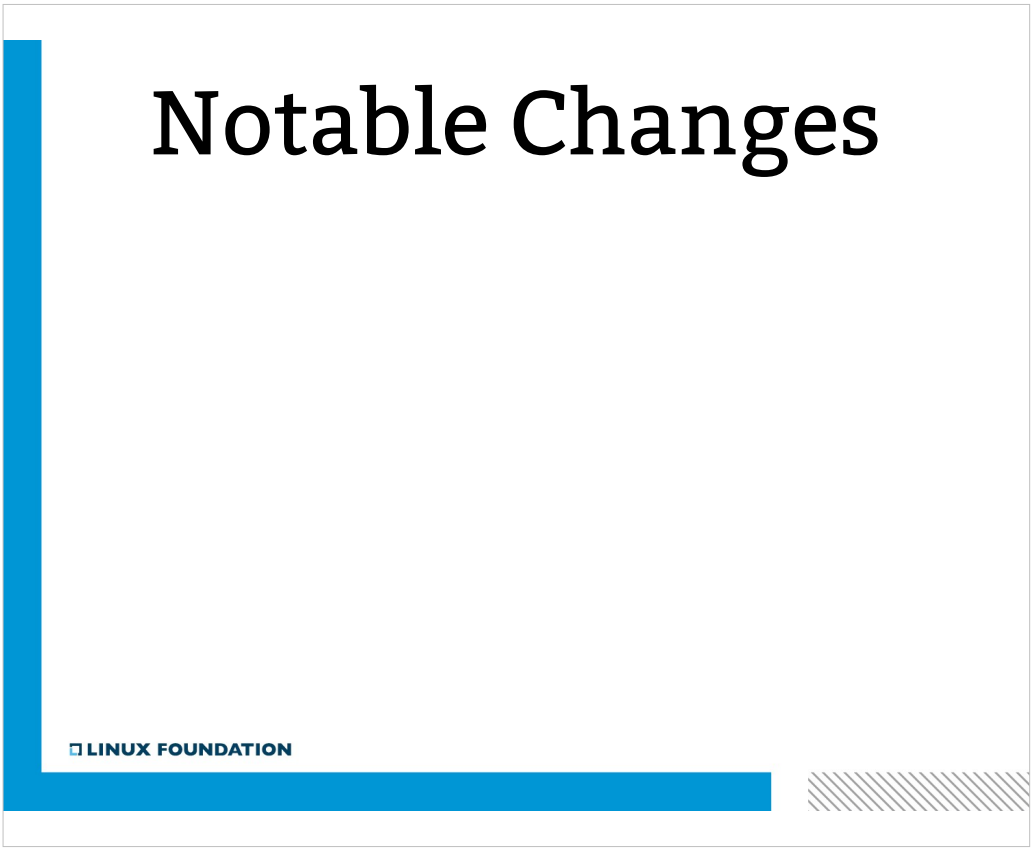
For example, the core kernel is only 5% of the code size, and 5% of the changes made were to the core kernel. Drivers make up 55% of the code, and 55% of the changes were to drivers.

Our rate of change is proportional across the whole kernel, this isn't just drivers that are changing.



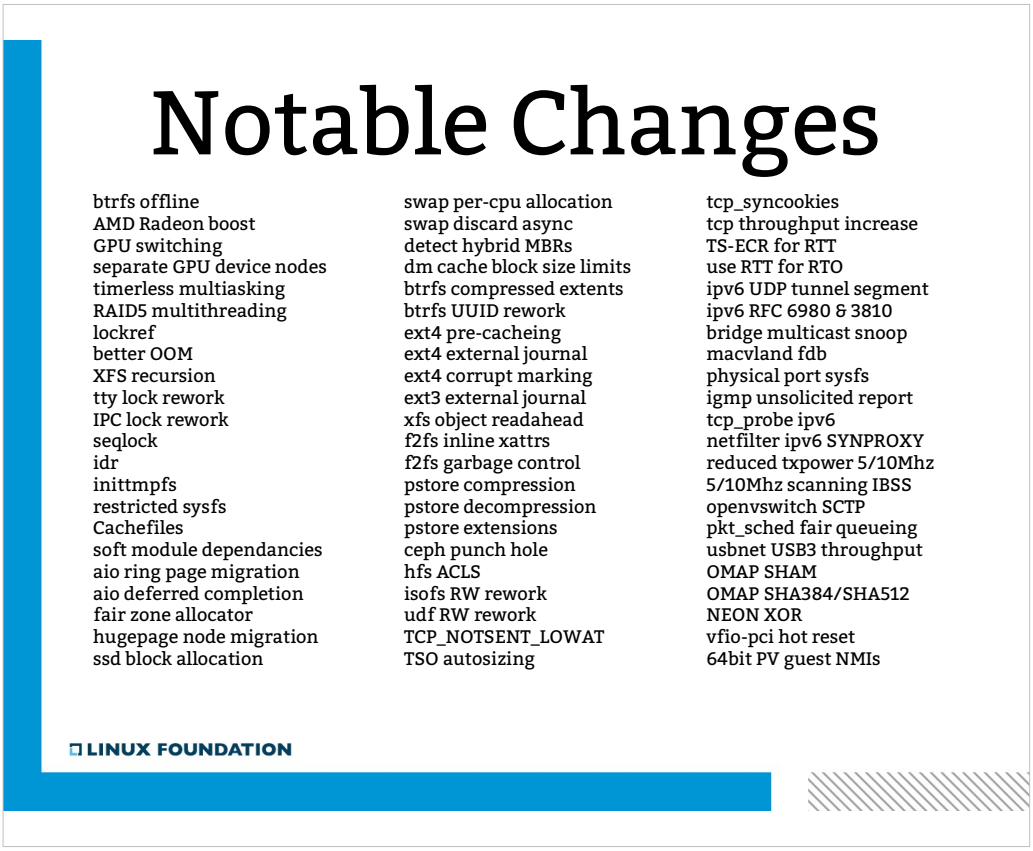
The 3.16 kernel release was the fastest we have ever created. This shows just how well the Linux kernel development model is working. We are growing in developers, companies, and how well we are accepting changes.

Note, this is just the number of patches that we have accepted, not all of the ones that have been actually submitted. Lots of patches are rejected, as anyone who has ever tried to submit a patch can attest to.



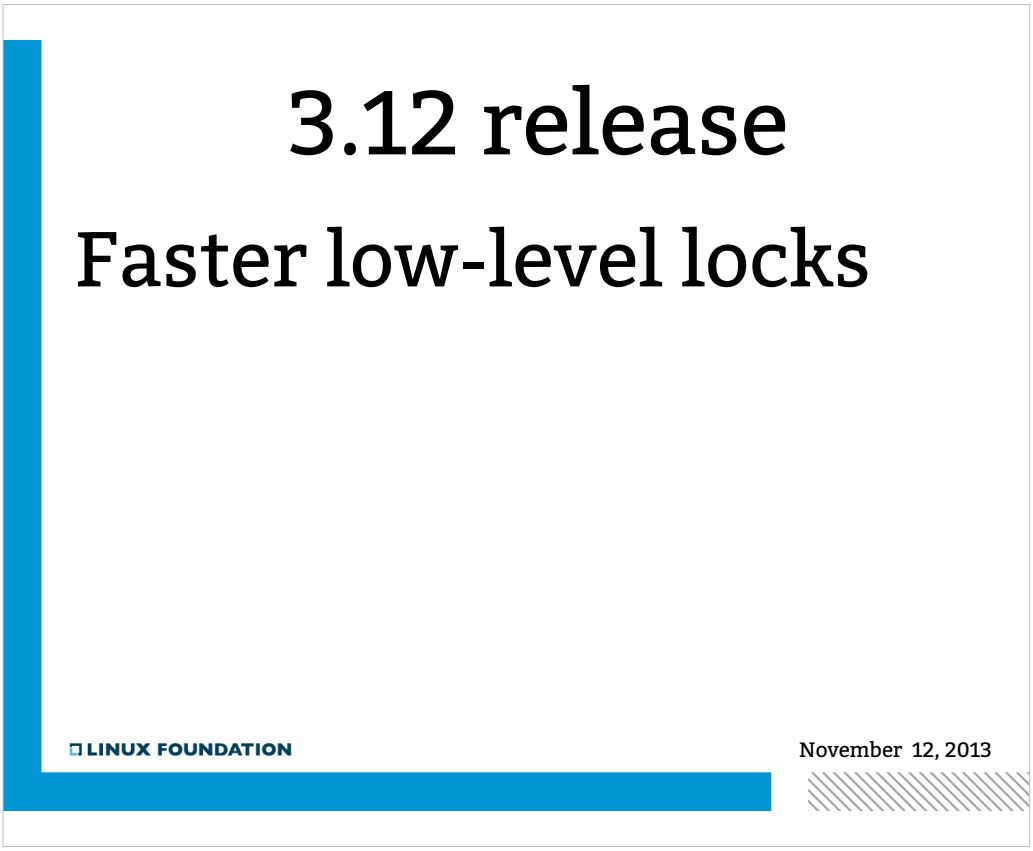
Let us look at some changes that have gone into the kernel over the past year.

I just want to highlight a few, there were over 92 thousand different changes during the past year, so here are just a few of them.



Here is just a tiny partial list of some of the changes that went into just the 3.12 kernel release. These aren't even getting into all of the driver changes that happened. To try to summarize a single kernel release is almost impossible. The kernelnewbiews site has a great list of everything that has changed if you want to know the details.

So let's just pick a very tiny number of feature that might be relevant to you in the releases this year.



The 3.12 kernel got a rewrite of the very low-level locks in the kernel. Linus did this work with some of the other core kernel developers. This is very unusual for an operating system. Once something like the basic locks are written, no one wants to ever touch them again, as it's one of the basic things that the whole kernel is based on.

This shows that the changes in Linux are at the very basic level at times, kernel developers are not afraid to revisit previously working code and make it better.

This also means if you are using an older kernel than 3.12, your machines could be running faster.

This feature lets your multiprocessor machine not have a timer tick constantly handled by a cpu just to determine that nothing else needs to be done. This can help with power management issues so that all processors can be properly shut down, as well as systems that do not have extra jobs other than the number of processors, allowing them to complete their tasks faster.

Again, a very low-level feature that lets you save power and go faster.

3.13 release

Multi-queue block layer

LINUX FOUNDATION

January 19, 2014

3.13 release

nftables

LINUX FOUNDATION

January 19, 2014

3.13 release

nfc payment support

LINUX FOUNDATION

January 19, 2014

3.14 release

Deadline scheduler

LINUX FOUNDATION

March 30, 2014

3.14 release

Kernel address space ranc

LINUX FOUNDATION

March 30, 2014

3.14 release

Antibufferbloat packet sc

LINUX FOUNDATION

March 30, 2014

3.15 release

Faster resume

LINUX FOUNDATION

June 8, 2014

3.15 release

Faster erase and zero portions of a file

LINUX FOUNDATION

June 8, 2014

3.15 release

File cross-rename

LINUX FOUNDATION

June 8, 2014

3.15 release

FUSE write improvements

LINUX FOUNDATION

June 8, 2014

3.16 release

Unified control group heirachy

LINUX FOUNDATION

August 3, 2014

3.16 release

TCP fast open on IPV6

LINUX FOUNDATION

August 3, 2014

3.16 release

CONFIG_USB_DEBUG rem

LINUX FOUNDATION

August 3, 2014

3.16 release

32bit VDSO on 64bit

LINUX FOUNDATION

August 3, 2014

3.17 release

File sealing - memfd

LINUX FOUNDATION

October 5, 2014

3.17 release

Thunderbolt on Apple hardware

LINUX FOUNDATION

October 5, 2014

3.17 release

getrandom()

LINUX FOUNDATION

October 5, 2014

3.17 release

Signed kexec kernels

LINUX FOUNDATION

October 5, 2014

3.18 release

unionfs

LINUX FOUNDATION

Sometime in 2014

3.18 release

bfp() syscall

3.18 release

Small network packet imp

LINUX FOUNDATION

Sometime in 2014

future release?

Live kernel patching

LINUX FOUNDATION

Sometime in 2014

future release?

kdbus

LINUX FOUNDATION

Sometime in 2014

future release?

kselftests

LINUX FOUNDATION

Sometime in 2014

future release?

O_BENEATH

LINUX FOUNDATION

Sometime in 2014

future release?

cgroup namespaces

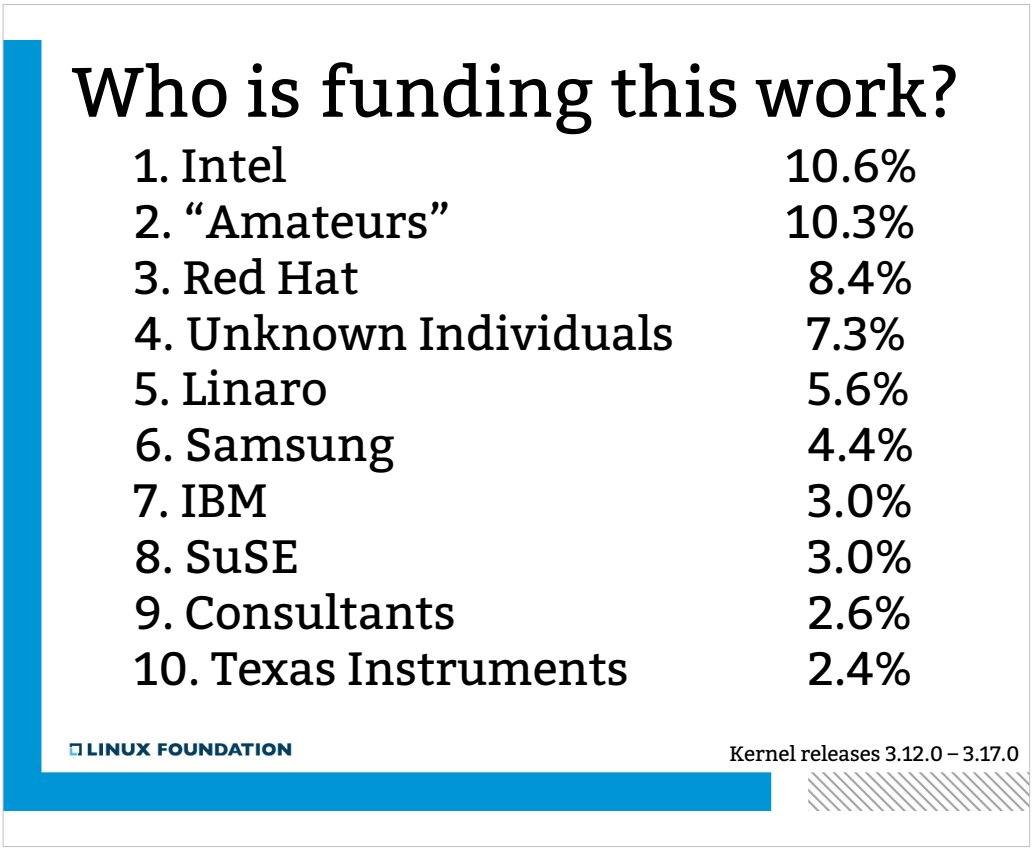
LINUX FOUNDATION

Sometime in 2014

future releases?

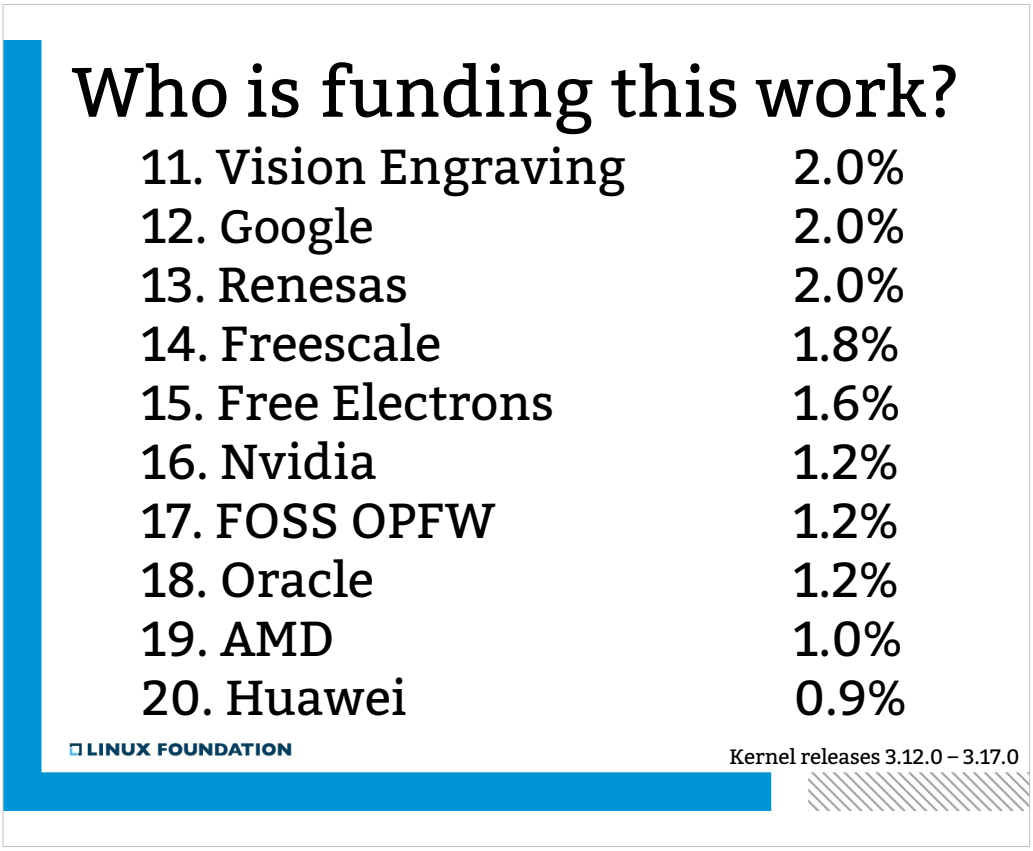
bfp() syscall

Sometime in 2014



So you can view this as either 17% is done by non-affiliated people, or 83% 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.



Vision Engraving is just one developer who has done 1519 patches last year. Google had about 1500 patches for the whole company.

FOSS Outreach Program for Women 900 patches. 20 women interns / students.

The appliation process for the next round of OPW just happened last month which resulted in 515 patches being accepted into the kernel tree for the 3.19 kernel release.


©¡WOUW!