



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

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

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
macvlan 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

3.13 release

nftables

3.14 release

Antibufferbloat packet scheduler

3.15 release

Faster resume

3.16 release

CONFIG_USB_DEBUG

3.16 release

32bit VDSO on 64bit

3.17 release

File sealing - memfd

3.18 release

unionfs

3.18 release

Major network speedup

future release?

Live kernel patching

future release?

kdbus

future release?

kselftests

future release?

O_BENEATH

future release?

cgroup namespaces

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%

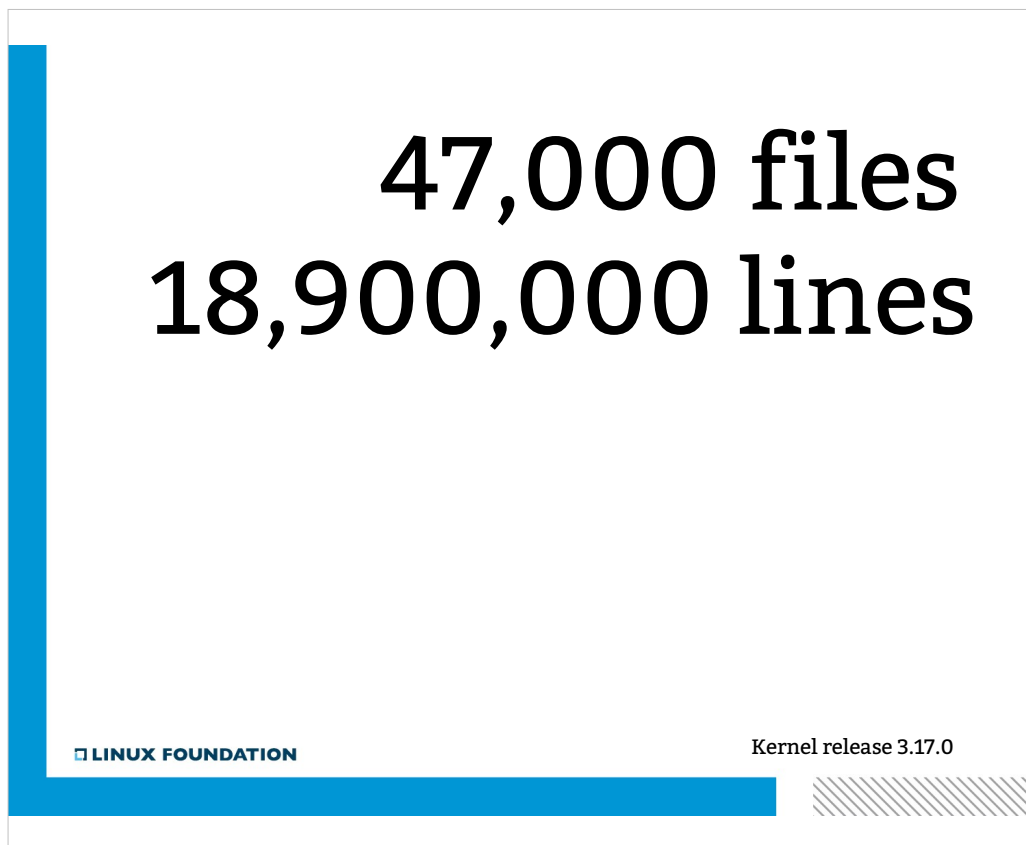




Last year in the Linux Kernel

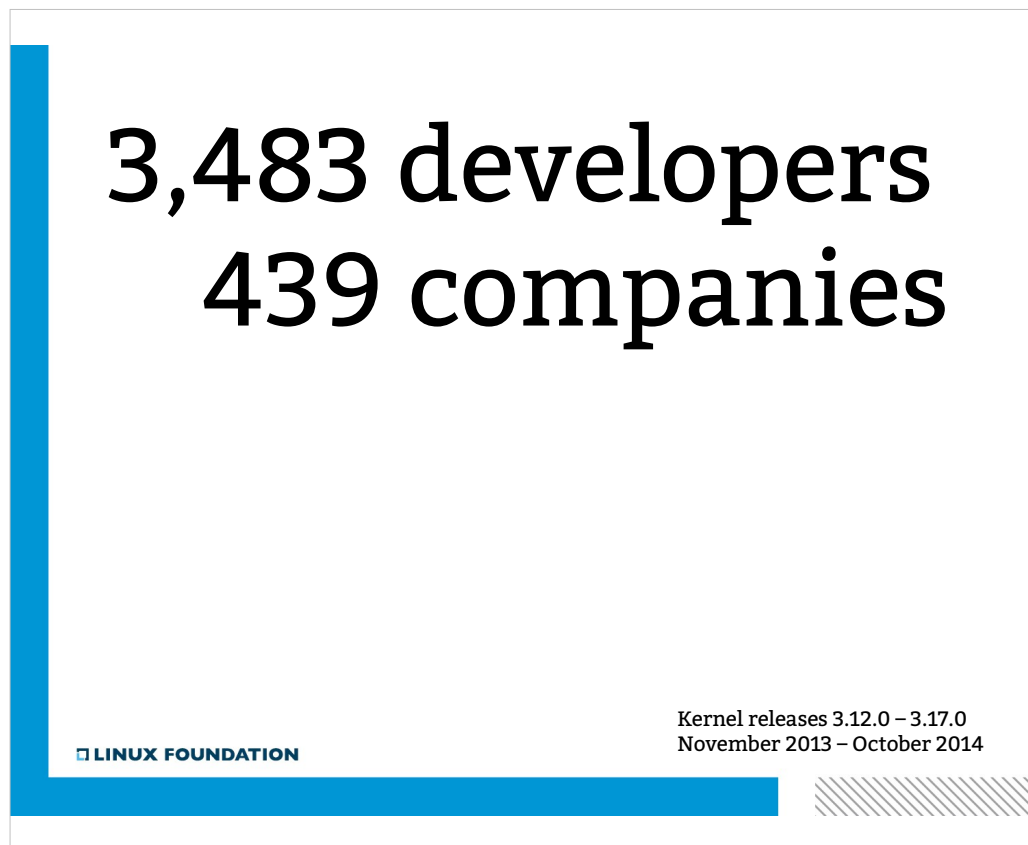
Greg Kroah-Hartman

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.

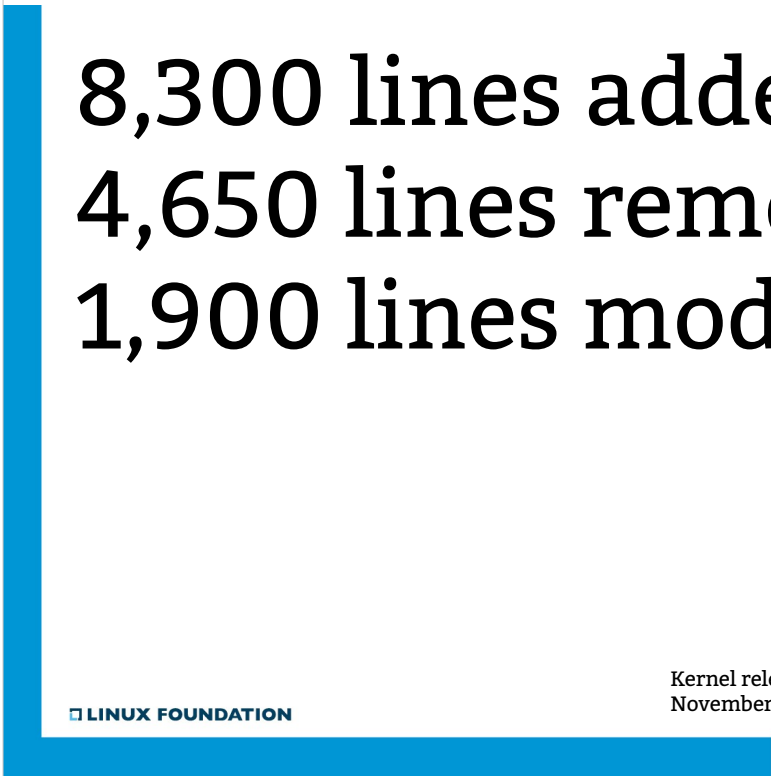


This is for the past year's worth of development, from the 3.12 through the 3.17 kernel release.

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.



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

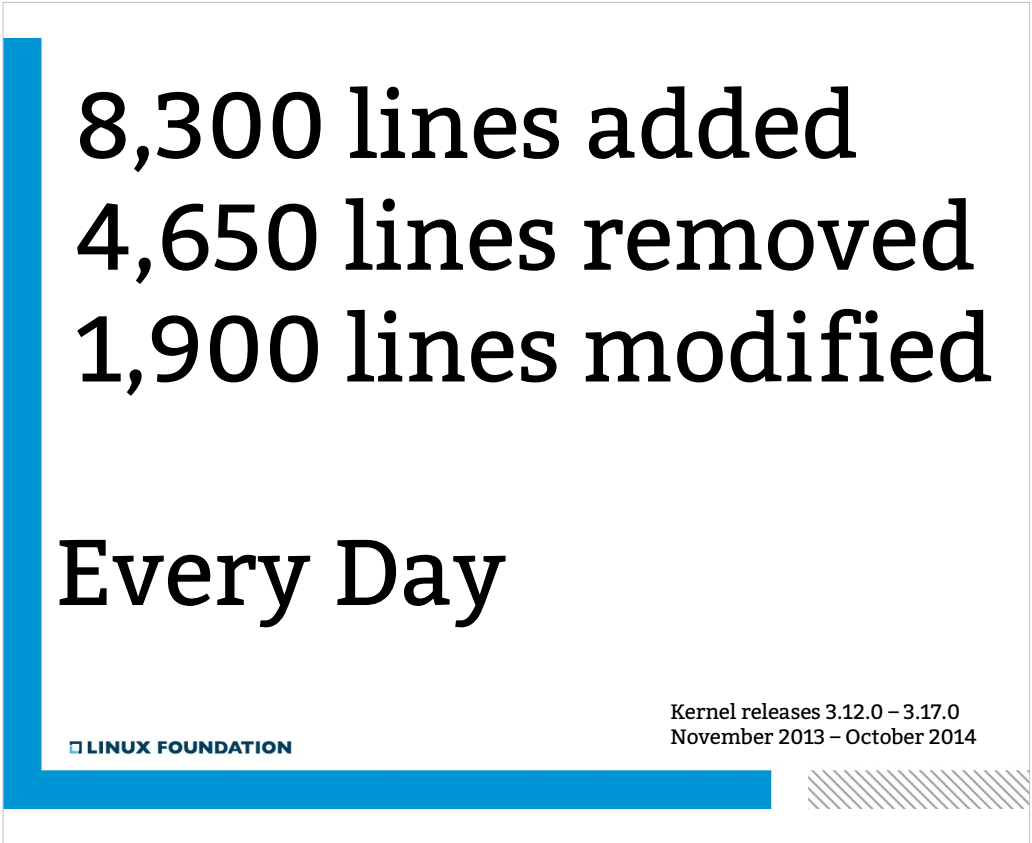
 LINUX FOUNDATION

Kernel releases 3.12.0 – 3.17.0
November 2013 – October 2014



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

Untill...



The infographic features a large blue L-shaped border on the left and bottom. The text is centered in a large, bold, black sans-serif font. At the bottom left, the Linux Foundation logo is present. At the bottom right, there is a small rectangular area with diagonal hatching.

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

Every Day

 LINUX FOUNDATION

Kernel releases 3.12.0 – 3.17.0
November 2013 – October 2014

You realize the unit of change.


This happens every day, and it keeps going up.



7.8 changes per hour

 **LINUX FOUNDATION**

Kernel releases 3.12.0 – 3.17.0
November 2013 – October 2014



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.



9.5 changes per hour

3.16 release

 LINUX FOUNDATION

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.

Notable Changes

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

 **LINUX FOUNDATION**

Here are just a few of the changes that went into the 3.12 kernel release, I didn't even start to look at the drivers. As over 92 thousand changes went into the kernel last year, it's impossible to try to summarize what has happened in any reasonable time.

So I just picked one or two different changes from each of the past kernel releases to show something that I thought was relevant.

If you want a good summary of the changes in a kernel release, look at the [kernelnewbies](http://kernelnewbies.org) web site.



3.12 release

Faster low-level locks

 LINUX FOUNDATION

November 12, 2013



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.

3.13 release

nftables

 LINUX FOUNDATION

January 19, 2014

The networking stack has changed how it defines packet handling again, replacing the iptables tool with something called nftables.

It provides backwards compatibility with iptables, and solves many of the problems that iptables had. It is userspace compiler that reads the rules, generates a virtual language and then sends that to the kernel to interpret.

3.14 release

Antibufferbloat packet scheduler

 LINUX FOUNDATION

March 30, 2014

Bufferbloat is a real problem for networking devices, and the kernel now has a scheduling algorithm that can properly solve the issue. Everyone should be turning it on in their systems.

3.15 release

Faster resume

 LINUX FOUNDATION

June 8, 2014

Some Intel engineers noticed that when resuming a system that was asleep, the hard disk can take the longest time as the kernel has to wait for the hardware to properly wake up before it can start talking to it.

The 3.15 kernel changed this, so that the kernel now goes off and does other things while the hard disk is waking up, greatly reducing the overall time that it takes the system overall to resume.

3.16 release

CONFIG_USB_DEBUG

 LINUX FOUNDATION

August 3, 2014

This is a configuration option that was removed in the 3.16 kernel. It took me about 2 years to remove this option.

Previously, if a user reported an issue with USB, the developers would ask them to rebuild their kernel with this configuration option enabled, run the test, and report the problem. That meant a lot of users just never did anything as they didn't know how to build their own kernel.

Now USB debugging is always built in (you can disable it entirely if you want to save space), and a user can dynamically enable it on a file, function, or even line number basis and report the result much easier, no kernel rebuild is needed.




3.16 release

32bit VDSO on 64bit

 LINUX FOUNDATION

August 3, 2014



There are some system calls, like the get-time-of-day call, that need to go really fast. This kernel change enabled these types of system calls to work properly when running a 64bit kernel and an old 32bit userspace program.

Normally this isn't a big deal, but it turns out that almost all games are 32bit programs, so this change greatly sped up all games running on Linux for users.

3.17 release

File sealing - memfd

 LINUX FOUNDATION

October 5, 2014

There is a new system call in 3.17 called memfd. This creates a memory space that can be “sealed” after you write data to it. You then pass that memory to another process and the process can “know” that the memory will not be modified by any other process.

This is used by a lot of graphics applications, and there are changes coming in the glib code to take advantage of it as well.

Future kernel features, like kdbus, also really want this feature.

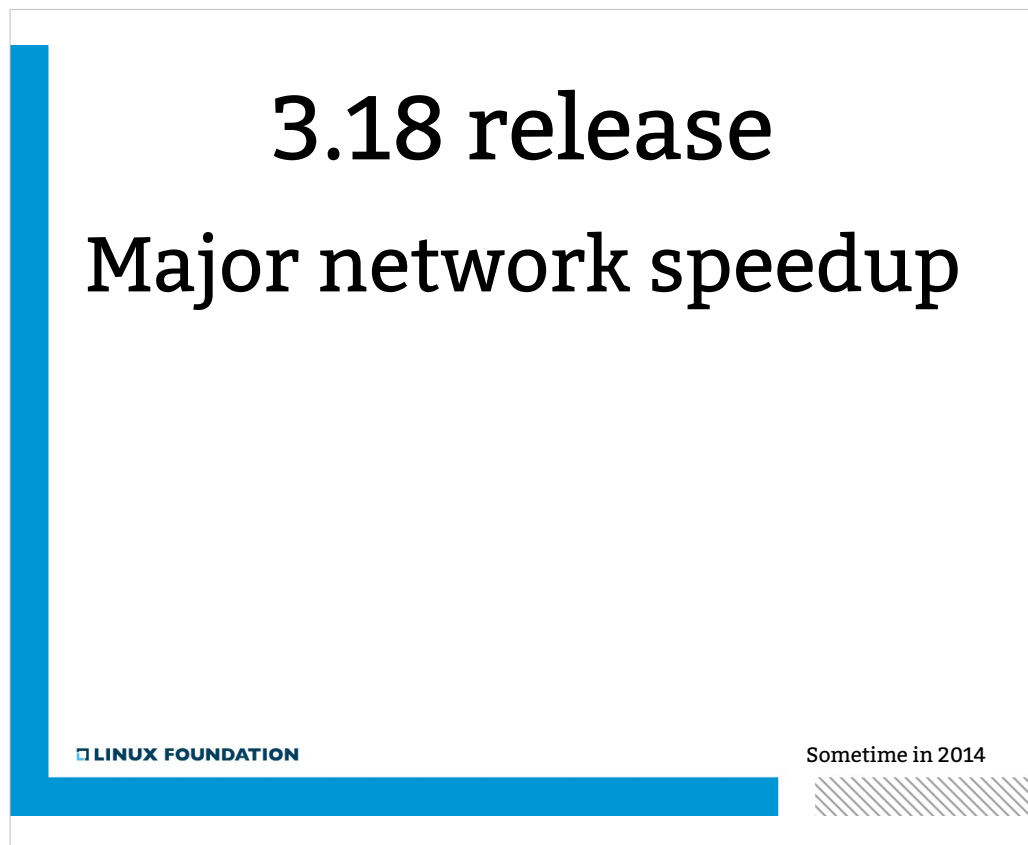
3.18 release unionfs

 LINUX FOUNDATION

Sometime in 2014



Yes, the 3.18 kernel will finally have unionfs. It's only taken about 10 years, but now Linux will have a valid stackable filesystem solution that is in the main kernel tree and does not require external patches.



One other notable change in the 3.18 kernel release will be a major speed up in some networking workloads.

When you have small packets, now the network stack can recognize this, and properly schedule them all together, taking advantage of the hardware. You can now get line-speed transfer rates using small packets, something that lots of people have been wanting for a while.

The idea for how to do this came from the block layer of the kernel, which had been doing this for small block writes for a number of years.

future release?

Live kernel patching

 LINUX FOUNDATION

Sometime in 2015

Let's guess at some of the features coming in future kernel versions.

One of the big ones that lots of people are watching is live kernel patching. This lets you modify your running kernel with a patch to resolve a security or bugfix, without requiring it to be rebooted.


There are two competing proposals out there, one from SuSE and one from RedHat. This past week saw a combined proposal from these developers on how to accomplish this, along with working kernel patches to do so.

future release?

kdbus

 LINUX FOUNDATION

Sometime in 2015



Kdbus is an IPC mechanism that uses a dbus-like protocol and puts it in the kernel, instead of doing it in userspace. This work has been happening for over a year now, and patches have been submitted for review a few weeks ago.


Based on that review, changes are happening, and a new set of patches should be submitted next week to address the issues found.

future release?

kselftests

 LINUX FOUNDATION

Sometime in 2015



The kernel has a built-in set of self tests (didn't know that, did you?) These tests build and run on the same machine that you built the kernel on.

The kernel testing maintainer has been working on those tests and modifying them so that you can install and run them on a target machine, which will make it easier to test embedded systems that sometimes can not run a compiler very easily.

future release?

O_BENEATH

 LINUX FOUNDATION

Sometime in 2015


O_BENEATH is a new file flag that is being proposed to make it illegal for any file that is passed to the kernel, to not traverse “backwards” in the file system. This is being asked for by people who want to make it easier to contain processes from running or accessing files outside of a specific location.

future release?

cgroup namespaces

 LINUX FOUNDATION

Sometime in 2015



There have been some patches proposed to make cgroups namespace aware. Right now cgroups are global, but people want to make them easier to virtualize and grant access to for a specific container.

I don't know if these patches will be accepted, but it's an interesting idea and might show up in a future kernel release.

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%

 LINUX FOUNDATION

Kernel releases 3.12.0 – 3.17.0

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.

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%

 LINUX FOUNDATION

Kernel releases 3.12.0 – 3.17.0

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 application 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.

