What's Up in the Land of the Linux Kernel
=quick orientation=
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td><a href="https://www.kernel.org/pub/">https://www.kernel.org/pub/</a></td>
</tr>
<tr>
<td>Git</td>
<td><a href="https://git.kernel.org/">https://git.kernel.org/</a></td>
</tr>
<tr>
<td>RSYNC</td>
<td>rsync://rsync.kernel.org/pub/</td>
</tr>
</tbody>
</table>

Latest Stable Kernel: 5.4.14

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>mainline:</td>
<td>5.5-rc7</td>
<td>2020-01-20</td>
</tr>
<tr>
<td>stable:</td>
<td>5.4.14</td>
<td>2020-01-23</td>
</tr>
<tr>
<td>longterm:</td>
<td>4.19.98</td>
<td>2020-01-23</td>
</tr>
<tr>
<td>longterm:</td>
<td>4.14.167</td>
<td>2020-01-23</td>
</tr>
<tr>
<td>longterm:</td>
<td>4.9.211</td>
<td>2020-01-23</td>
</tr>
<tr>
<td>longterm:</td>
<td>4.4.211</td>
<td>2020-01-23</td>
</tr>
<tr>
<td>longterm:</td>
<td>3.16.81</td>
<td>2020-01-11</td>
</tr>
<tr>
<td>linux-next:</td>
<td>next-20200124</td>
<td>2020-01-24</td>
</tr>
</tbody>
</table>
Linux 5.4 is the latest kernel line

*an it's a "longterm kernel": supported at least 2 years (likely 6)*
5.5 under development

release expected on February 3th (+/- one week)
5.6 expected mid-april
likely to late for Ubuntu 20.04 LTS
maybe to late for Fedora 32
new releases every 9 or 10 weeks
"well tuned engine"
each ~13.500 changes
on average +300.000 lines each
recent happenings
last year at this point
I showed this slide:
a few hiccups recently

so much for "well tuned"...
[flashback from 2019]

ZoL breakage, big performance problem, block corruption bug, Code of Conduct & Linus break
no similar disturbances of the force recently :-}
CoC is nothing much heard of

incident reports now regularly on
www.kernel.org/category/about.html

(there were 5 reports in 2019)
forgot something
there was some disturbance
a LPC talk by
Dmitry Vyukov
Developer satisfaction?

communication style :

lost patches :

feeling non-productive :

struggling with tools :

lost of patch versions :

lost of "nitpicks" :

Do I want to send a patch that I don't have to?...

duplicate work :

lost bugs :

Do I want to finish my patch?...

introducing regressions :

what's the status of my patch? :

can't add tests :

late reverts :

inconsistency :

non-transparency :(
pointed out many problems in current kernel development approach

lwn.net/Articles/799134/ (links to slides)
www.youtube.com/watch?v=iAfrrNdl2f4
Workflow group started
goal: improving & defragmenting kernel development workflow

https://lwn.net/Articles/803619/
a lot of work ahead...

will take years, remains to be seen what comes out of it
Welcome #Gerrit changes for #linux kernel:
linux-review.googlesource.com/c/virt/kvm/kvm...

and the mailing list version for contrast:
lore.kernel.org/lkml/202001231...

Gerrit has side-by-side diffs, full expandable context, non-lossy comments attached to lines. Here are docs:
linux.googlesource.com/Documentation/...
and no:

kernel development won't move to gitlab or github ;-)
forgot something else
there was a bit of a stir
Linus: "Don't use ZFS. It's that simple."

https://www.realworldtech.com/forum/?threadid=189711&curpostid=189841

note! he meant "ZFS from Oracle" there, not OpenZFS or ZOL!
see https://www.realworldtech.com/forum/?threadid=189711&curpostid=189959
no hope for a ZFS merge
expected with the well known licensing issues

good to hear from him
even if the licensing gets resolved
likely other potential problems...

patents (see NetApp vs Sun/Oracle)
code quality/interaction collides with Linux kernel code quality standards
I agree with him:
better stay away from it
(reevaluate if the situation changes)
and *maybe* investing in Btrfs would be the better approach anyway
=major happenings=
Wireguard support expected in Linux 5.6
promising, much praised
VPN tunnel solution

even Linus praised it
stateless, easy configuration, quick reconnect, very elegant Linux integration, ...
a crypto API for wg:

showstopper ZINC finally resolved in 5.5

"frankenzinc"
=major happenings=
Realtime support soon to get mainlined

RT-Tree/PREEMT_RT

(TBH: said that last year here, too, but wait for it….)
aka

let Linux control your Laser cutter ;-) important for industry usage, among others
CONFIG_PREEMPT_RT
already in mainline (5.3)

deps missing, so not exposed
and thus not usable
biggest missing part

reworking printk()

https://lwn.net/Articles/800946/
5.7 seems earliest possible target right now
RT is about to make it to mainline. Now what?

Steven Rostedt

Embedded Recipes

https://www.youtube.com/watch?v=sZMNm1o7FQw
Realtime

reminder:

real time does not mean faster, it (in the end) means: be predictable

(enabling PREEMT_RT often lowers throughput a bit; that's why it's unlikely to get enabled in default kernels from Fedora, Ubuntu, … )
=major happenings=
BPF (aka eBPF) changing the Linux kernel drastically one step at a time – and most people don't notice it
BPF is a programmable VM in the kernel

VM like a Java VM
BPF for now mostly shows itself here:

TC

XDP (network fastpath)

BCC & bpftrace
The following eBPF components are fully supported in RHEL 8.2:

- The BPF Compiler Collection (BCC) tools package is a userspace collection of dynamic kernel tracing utilities that use the eBPF virtual machine for creating efficient kernel tracing and manipulation programs. BCC provides tools for I/O analysis, networking, and monitoring of Linux operating systems using eBPF.

- The BCC library which allows the development of tools similar to those provided in the BCC tools package.

- The eBPF for tc feature enables programmable packet processing inside the kernel network data path.

The following eBPF components are currently available as a Technology Preview:

- The `bpftrace` tracing language

- The eXpress Data Path (XDP) feature

For details regarding the Technology Preview components, see Section 6.3.
BPF Performance Tools (book)

This is the official site for the book BPF Performance Tools: Linux System and Application Observability, published by Addison Wesley (2019). This book can help you get the most out of your systems and applications, helping you improve performance, reduce costs, and solve software issues. Here I'll describe the book, link to related content, and list errata and updates.

The book is available on Amazon.com (paperback, kindle), InformIT (paperback, PDF, etc), and Safari (here and here). The paper book was released in December 2019 but sold out immediately; more copies printed soon. ISBN-13: 9780136554820.

The Amazon Kindle preview shows the first 100 pages out of this 880 page book.

As an example new tool from the book, readahead.bt provides a new view of file system read ahead performance: the age of read-ahead pages when they are finally referenced, and unused read-ahead pages while tracing:

```
# readahead.bt
Attaching 5 probes...
^C
Readahead unused pages: 128

Readahead used page age (ms):
@age_ms:
[1] 2455
[2, 4] 8424
[4, 8] 4417
[8, 16] 7680
[16, 32] 4352
[32, 64] 0
[64, 128] 0
[128, 256] 384
```

The book covers many of the existing tools as well, for example, tcplife for efficiently logging TCP
a lot of small improvements in BPF land promise even more

happened or happening currently
• BPF STRUCT_OPS (5.6)
• BPF Trampolines (5.5)
• Revolutionize bpf tracing and bpf C programming (5.5)
• Compile Once, Run Everywhere/CO-RE (5.4)
• Bounded Loops (5.3)
• BTF type info for vmlinux (5.2)
• Global Data (5.2)
• BPF Verifier Scalability (5.2)
• Spinlocks (5.1)
not going into details here, but…
together they are building blocks for a quite different Linux kernel
talks at devconf.cz that have "BPF" in the description:

· What is eating my traffic? Packet thieves revealed
· Open vSwitch with AF_XDP as a userspace datapath
· strace: fight for performance
· An introduction to bpftrace tracing language
· Using bpftrace with Performance Co-Pilot & Grafana
· bpftrace internals
· Generate seccomp profiles for containers using bpf
· Traceloop: Tracing containers syscalls using BPF
· Checking your cryptography usage with eBPF
· BPF: The Status of BTF, producers, consumers.
· Using systemd features to secure services
might turn out to revolutionize Linux

remains to be seen how things evolve
Bpfilter (and user-mode blobs) for 4.18

By Jonathan Corbet
May 30, 2018

In February, the `bpfilter` mechanism was first posted to the mailing lists. Bpfilter is meant to be a replacement for the current in-kernel firewall/packet-filtering code. It provides little functionality itself; instead, it creates a set of hooks that can run BPF programs to make the packet-filtering decisions. A version of that patch set has been merged into the net-next tree for 4.18. It will not be replacing any existing packet filters in its current form, but it does feature a significant change to one of its more controversial features: the new user-mode helper mechanism.

[...]

The replacement of netfilter, even if it happens as expected, will take years to play out, but we may see a number of interesting uses of the new user-mode helper mechanism before then. The kernel has just gained a way to easily sandbox code that is carrying out complex tasks and which does not need to be running in a privileged mode; it doesn't take much effort to think of other settings where this ability could be used to isolate scary code. Just be careful not to call the result a "microkernel" or people might get upset.
Toke Høiland-Jørgensen @toke_dk · Dec 14, 2019
Another step on the path towards Linux becoming a BPF-powered microkernel? Fascinating to watch!

Brendan Gregg @brendangregg · Dec 14, 2019
Facebook's Martin KaFai Lau has developed BPF STRUCT_OPS to allow implementing tcp_congestion_ops (and more) in BPF. marc.info/?l=linux-netde...
closing side note, as EOT:
sadly, Realtime and BPF
do not work together :-/
#yet
"Is that guy standing in front of us a reliable source?"
not a kernel developer
Die Sicherheits-Checkliste
Handy, Router, NAS, Smart-TV, Server, Browser, Facebook ...
Kernel regression tracking, part 2

By Jonathan Corbet
November 6, 2017
2017 Maintainers Summit

The tracking of kernel regressions was discussed at the 2017 Kernel Summit; the topic made a second appearance at the first-ever Maintainers Summit two days later. This session was partly a repeat of what came before for the benefit of those (including Linus Torvalds) who weren’t at the first discussion, but some new ground was covered as well.

Thorsten Leemhuis started with a reprise of the Kernel Summit discussion, noting that he has been doing regression tracking for the last year and has found it to be rather harder than he had expected. The core of the problem, he said, is that nobody tells him anything about outstanding regressions or the progress that has been made in fixing them, forcing him to dig through the lists to discover that information on his own. He had, though, come to a few conclusions on how he wants to proceed.

First, he will try again to establish the use of special tags to identify regressions. His first attempt had failed to gain traction, but he agreed that he perhaps had not tried hard enough to publicize the scheme and get developers to use it. He will be looking into using the kernel Bugzilla again, even though it still seems like unpleasant work to him. He’ll try to improve the documentation of how regressions should be tracked and handled. There is a plan to create a new mailing list on vger.kernel.org, with the idea that regression reports would be copied there. He will put more effort into poking maintainers about open regressions.

The discussion quickly turned to the problem (as seen by some) of the many kernel subsystems that do not use the kernel.org Bugzilla instance for tracking bugs. Peter Anvin said that many developers don’t see much value in that system. Reported bugs tend to say something like “my laptop doesn’t boot” with no further information; that tends not to be useful for the identification of any actual bugs. Beyond that, many bugs reported against the core kernel or x86 architecture turn out to be driver bugs in the end.

Users, it was suggested, should be explicitly directed to the mailing lists when reporting bugs for the subsystems that do not use Bugzilla. Laura Abbott said...
Thorsten 'the Linux kernel logger' Leemhuis

@kernellogger

#Linux #kernel and related areas like #xorg, #mesa, #wayland, #qemu, and #mdadm. Account 6/6, for the others see leemhuis.info/me/ Opinions are my own.

Hannover, Germany  Joined November 2009

167 Following  1,480 Followers

Util- #Linux 2.35 is out:

"Release highlights: [...] kill(1) now uses pidfd kernel feature to implement a new command-line option
--timeout [...] scriptlive(1) this NEW COMMAND re-execute stdin log by a shell in PTY session [ 1]"
Kernel Vanilla Repositories

Package repository with Linux vanilla kernels for Fedora

The Fedora kernel vanilla repositories offer various RPM packages that contain vanilla builds of different Linux kernel version lines. These packages are meant for Fedora users that want to access the latest stable or pre-releases of Linux quickly and comfortably. In
questions on the topics covered so far?

#status, #COC, #workflow, #ZFS, #realtime, #BPF
=second part of the talk=}
recent changes & WIP that are
(1) useful/helpful
day to day
or
(2) good/important
to know
useful/helpful day to day
btrfs, ext4 and xfs: how many of these file systems support deduplication with COW?
COW == Copy-on-write
copy big files or directory trees in a fraction of a second
aka "cp --reflink" – useful for
snapshotting, provisioning, container images (sharing files), ...
raise your hands...
if you are listening ;-)
raise your hands...
if you have heard about deduplication with Copy-on-Write (COW)
raise your hands… if you think *Btrfs* supports it
raise your hands…
if you think *XFS*
supports it
raise your hands…
if you think *Ext4*
supports it
Btrfs (and ZFS) support it since the beginning, ext4 does not
xfs learned it quite recently (4.1x/5.x – "reflink feature")
default and production ready since xfsprogs 5.1
XFS - Data Block Sharing (Reflink)

Matt Keenan
PRINCIPAL SOFTWARE ENGINEER

Following on from his recent blog XFS - 2019 Development Retrospective, XFS upstream maintainer Darrick Wong dives a little deeper into the Reflinks implementation for XFS in the mainline Linux Kernel.

Three years ago, I introduced to XFS a new experimental "reflink" feature that enables users to share data blocks between files. With this feature, users gain the ability to make fast snapshots of VM images and directory trees; and deduplicate file data for more efficient use of storage hardware. Copy on write is used when necessary to keep file contents intact, but XFS otherwise continues to use direct overwrites to keep metadata overhead low. The filesystem automatically creates speculative preallocations...

XFS now supports shared copy-on-write data extents

The XFS file system supports shared copy-on-write data extent functionality. This feature enables two or more files to share a common set of data blocks. When either of the files sharing common blocks changes, XFS breaks the link to common blocks and creates a new file. This is similar to the copy-on-write (COW) functionality found in other file systems.

Shared copy-on-write data extents are:

**Fast**
Creating shared copies does not utilize disk I/O.

**Space-efficient**
Shared blocks do not consume additional disk space.

**Transparent**
Files sharing common blocks act like regular files.
guess it will take years till "XFS supports COW" makes it around

similar to how "LVM snapshots are very slow" is still in quite some heads

(which is not the case for many years now if you are doing it properly) #sigh
online fschk nearly stable, too
XFS - Online Filesystem Checking

Matt Keenan
PRINCIPAL SOFTWARE ENGINEER

XFS Upstream maintainer Darrick Wong provides another instalment, this time focusing on how to facilitate sysadmins in maintaining healthy filesystems.

Since Linux 4.17, I have been working on an online filesystem checking feature for XFS. As I mentioned in the previous update, the online fsck tool (named xfs scrub) walks all internal filesystem...
useful/helpful day to day
exFAT support (5.4)
easily mount the sd cards from your modern cameras
not enabled in Fedora right now
interim solution in staging tree

*better implementation (sdfat)*

*ready soon?*
useful/helpful day to day
virtio-fs (5.4)
more efficient way to share host files or directories with a VM
similar to 9p, but promises to deliver way better performance
partly based on code from FUSE
see virtio-fs.gitlab.io for details
qemu support still WIP :/-
useful & good to know
io_uring matures (5.x)
finally brought proper, Asynchronous I/O (AIO) to Linux last year
easy to use, too
at least compared to older Linux AIO
AIO kinda standard in the Windows world

kinda since forever
AIO unusual in the Linux world, as the kernel support wasn't the best
io_uring in the past few months got lot of finetuning and was much praised
SSDs are getting crazy performance. We so need async IO to overcome the syscall overhead.

https://twitter.com/mjpt777/status/1215209572681515008
consider it, when writing new code that really needs I/O performance

*high throughput and/or low latencies*
Faster IO through io_uring

Jens Axboe

The rapid growth of io_uring
[LWN subscriber-only content]

By Jonathan Corbet
January 24, 2020

One year ago, the io_uring subsystem did not exist in the mainline kernel; it showed up in the 5.1 release in May 2019. At its core, io_uring is a mechanism for performing asynchronous I/O, but it has been steadily growing beyond that use case and adding new capabilities. Herein we catch up with the current state of io_uring, where it is headed, and an interesting question or two that will come up along the way.

Classic Unix I/O is inherently synchronous. As far as an application is concerned, an operation is complete once a system call like read() or write() returns, even if some processing may continue behind its back. There is no way to launch an operation asynchronously and wait for its completion at some future time — a feature that many...
operation until some previous operation has completed. What is rather more difficult is moving information between operations. In Metzmacher's case, he would like to call openat() asynchronously, then submit I/O operations on the resulting file descriptor without waiting for the open to complete.

It turns out that there is a plan for this: inevitably it calls for … wait for it … using BPF to make the connection from one operation to the next. The ability to run bits of code in the kernel at appropriate places in a chain of asynchronous operations would clearly open up a number of interesting new possibilities. "There's a lot of potential,"
useful/helpful day to day
Blkiocost (5.4)
(formerly called Io.Weight)
better, more flexible
IO-controller
among others

limits only when there is contention
it's a cgroup2 controller
side note:
move to cgroup2 finally happening
after many years of preparation…
Fedora 31 & RHEL8 have it
others will follow
Sunday, January 26
1:30pm - 2:25pm

Brave new world of unified cgroup hierarchy
useful/helpful day to day
USB4 support slated for 5.6
raise your hands: who cares?
all of you should care!
USB4 not that far away

Intels Tiger Lake (Core i-11xxx?) will have it, expected this Summer
5.6 likely be to late for Ubuntu 20.04

so in two years from now, when USB4 might be pretty standard, your 20.04 usb key might not boot, as its kernel won't support USB4 unless backported or using (20.04.$((y>=2) && flavour != server))
useful/helpful day to day
support for AMD Radeon Rx 5700 (Navi10)

Linux 5.3 & Mesa 19.2
support for AMD Radeon Rx 5500 (Navi14)

Linux 5.4 & mesa 19.2
all recent Radeon:
mesa 20.0 to support OpenGL 4.6
thx to radeonsi (part of the amdgpu family)
mesa 20 expected soon
shows AMD gets a lot right these days
= good to know =
WIP to support Intel Gen12 graphics (Tiger Lake & Xe)

*kernel and mesa*
i965 and Iris support OpenGL 4.6, too

(since Mesa 19.3)
Iris (Gallium3D based) might soon take over
IOW: AMD & Intel have quite good FLOSS driver stacks these days (but not perfect)  
my 2 cent: if you can, avoid Nvidia
= good to know =
Nouveau (FLOSS driver for GeForce) still leaves a lot to wish for :-/
no 3D with latest-gen (Turing)

no distributable Firmware from Nvidia yet :-/

but looks like that is finally about to change soon...
Nouveau & all recent GPUs:
unable to reach high speed or low power states

distributable Firmware lacks support
Open Source, Linux Kernel, and NVIDIA [S21398]

We'll report up-to-the-minute developments on NVIDIA's status and activities, and possibly depending on last-minute developments a few future plans and directions, regarding our contributions to Linux kernel; supporting Nouveau (the open source kernel driver for NVIDIA GPUs, that is in the Linux kernel), including signed firmware behavior, documentation, and patches; and NVIDIA kernel drivers.

John Hubbard, Principle SW Engineer, NVIDIA

Primary Topic: Supercomputing
Industry Segment: Software
Topics: Frameworks [DL and non-DL] / Libraries / Runtimes
a new hope?

Nvidia getting more open?

let's wait and see…
= good to know =
clang can build the kernel now on x86-64
might still be a bit bumpy…
= good to know =
PSI Monitor (5.1)

enhanced Pressure-stall Information (PSI), which was added in 4.20
PSI == /proc/loadavg on steroids
cpu, mem and i/o separately and more telling
[thl@thl ~]$ head /proc/pressure/*

```bash
===> /proc/pressure/cpu 
===> /proc/pressure/io 
===> /proc/pressure/memory 

[thl@thl ~]$  
```
PSI Monitor allows better OOM handling

prevent desktop shell from stalling
four solutions in the works

Android has something

PSI Monitor – from Endless

oomd – from Facebook for Systemd

EarlyOOM – from the Gnome camp

(the latter proposed for Fedora 32; big discussion recently)

https://pagure.io/fesco/issue/2320
= good to know =
bcachefs – not much progress recently

aims to compete with btrfs and ZFS
will take years till ready, if ever

right now mostly a one man show
but has a loud fan crowd already...
currently it looks like:

btrfs is better suited

(and not as bad as its reputation)

but yeah, still leaves a few things to wish for
= good to know =
Multipath TCP
will be in 5.6

(merged to linux-next yesterday)
Linux is the first OS to support MPTCP v1, please give me that twitter dopamine. Lots... git.kernel.org/pub/scm/linux/...

3:29 PM · Jan 24, 2020 · Twitter Web App

83 Retweets 217 Likes

Dmitry Vyukov @dvukov · 13h
Replies to @davem_dokebi
My skewed view on MPTCP: it’s better than any other networking protocol b/c smb bothered to add syzkaller descriptions year before it’s upstreammed :)
github.com/google/syzkaller
(That smb even bothered to create github account just for that! Was it
=useful/helpful day to day=
support for Raspberry Pi 4

Linux 5.5/5.6
support the popular SBS out of the box in Fedora et. al.
= there is more="
security
Lsm stacking
Lockdown (5.4)
y2038
Kunit
glibc wrapping
pidfd
livepatching
RISC-V
5 level paging
retpoline optimizations
asynchronous CPU scheduling
core scheduling & address space isolation (WIP)
many other optimizations (compaction, Cache line movement/TLB misses)
400GBit ethernet
VirtualBox shared folder (5.6)
nfs cross server copies (5.5/5.6),
cifsroot (5.5)
thinkpad privacyGuard (5.4)
zoned block device (5.6?)

WSL uses Linux
= Takeaways =
takeaways

fine tuned engine, running fast as ever
takeaways

CoC was nothing much heard of this year
takeaways

Wireguard approaching
takeaways

Realtime approaching
takeaways

XFS now supports deduplication
takeaways

for fast I/O, consider io_uring
takeaways

lot's of great improvements

and better hardware support, too
BPF really changes the Linux kernel
wanna know more? ask, or use a search engine, Luke!
social media madness ;-)

https://lwn.net/
takeaways

follow me

@kernellogger on twitter or
kernellogger@fc.leemhuis.info on friendi.ca
(the latter works on mastodon and diaspora, too)
Thorsten 'Das Leben, das U...@thleemhuis Follows you
Das Leben, das Universum und der ganz Rest. Account 1/6, für die anderen siehe leemhuis.info/me/ Ansichten sind meine eigenen.

Thorsten 'Gnome & Flatpak'@knurd42 Follows you
The #gnome and #flatpak personality of @knurd42. Account 4/6, for the others see leemhuis.info/me/ Opinions are my own.

Thorsten 'FOSS & Life, The ...@knurd42 Follows you
Free & Open source software as well as Life, the universe, and everything. Account 3/6, for the others see leemhuis.info/me/ Opinions are my own.

Thorsten 'the Linux kernel I...@kernellogger
#Linux #kernel and related areas like xorg, mesa, wayland, qemu, and systemd. Account 6/6, for the others see leemhuis.info/me/ Opinions are...

Thorsten 'Computer & freie ...@thleemhuisfoss Follows you
Computerkram & Free and Open Source Software des @thleemhuis. Account 2/6, für die anderen siehe leemhuis.info/me/ Ansichten sind...

Thorsten 'Red Hat, Fedora ...@knurd42rhlc Follows you
The #RedHat, #Fedora, and #CentOS personality of @knurd42. Account 5/6, für die anderen siehe leemhuis.info/me/ Opinions are my own.
feedback

one more thing

please provide feedback

talk to me: negative and positive feedback welcomed
This talk gives an overview of recent and current developments in the Linux kernel; it will discuss what major changes the latest kernel versions brought, what the next version will bring, and what the Linux developers are working on for future releases. In that scope the presentation will also discuss changes in software close to the kernel; that for example includes Mesa, as its 3D drivers and work hand in hand with the graphics drivers in the Linux kernel.

Speakers

Thorsten Leemhuis
Technical Writer & Kernel regression tracker, heise Medien, c't
Thorsten works as editor for Heise Medien, which publishes the German c't magazine and runs the tech news site heise.de. For both he writes a column called "Kernel Log", which regularly discusses developments in the Linux kernel and areas close to it. Thorsten also was a major Fedora... Read More →

Sunday January 27, 2019 9:00am - 9:50am
D0206
Kernel, Beginner - no experience needed

What did you think?
😊 😞 😞
that's it – questions?

(TWIMC: this is slide #183)
mail: linux@leemhuis.info, thl@ct.de
GPG Key: 0x72B6E6EF4C583D2D
social media: @kernellogger, @knurd42rhfc on #twitter & #friendica;
four others, see www.leemhuis.info/me/

#EOF