What's new in the Linux kernel

and what's missing in Debian

Ben Hutchings · DebConf 23

Ben Hutchings

- Working on Linux kernel and related code for Debian and in paid jobs for about 15 years
- Debian kernel and LTS team member, doing various kernel packaging and backporting work
- Formerly maintained Linux long-term stable branches needed by Debian

mainline:	6.6-rc1	2023-09-10
stable:	6.5.3	2023-09-13
stable:	6.4.16 [EOL]	2023-09-13
longterm:	6.1.53	2023-09-13
longterm:	5.15.131	2023-09-06
longterm:	5.10.194	2023-09-02
longterm:	5.4.256	2023-09-02
longterm:	4.19.294	2023-09-02
longterm:	4.14.325	2023-09-02
linux-next:	next-20230913	2023-09-13

Linux releases early and often

- Linux has feature releases about 5 times a year, plus stable updates every week or two
- Some features aren't really ready or complete in their first kernel release
- Some will need changes elsewhere to be useful:
  - New user-space tool to configure it
  - New version of existing user-space tool
  - Applications and libraries using new API
  - Packaging or infrastructure changes
- I'll talk about new features in Linux 5.19 to 6.5 inclusive

Recap of previous years' features (1)

Added support for:

- ACPI
- Hibernation
- Relocatable kernel image (PIE)

Architecture extensions including:

- Vector extension (variable-length SIMD)
- 64 kiB TLB entries

Recap of previous years' features (2)

io_uring

- Multi-shot `accept`
- Extended attribute (xattr) operations
- Zero-copy network transmit
- Parallel direct I/O (on some filesystems)
- Multi-shot timers
- Rings in user-space memory

Recap of previous years' features (3)

ID-mapped mounts

Now supported by more filesystems:

- overlayfs
- squashfs
- tmpfs

Can be used by:

- crun, LXC, and other container software
- `mount` (`X-mount.idmap` option)
- systemd

User-space block drivers [6.0]

- **ublk** block driver in kernel delegates to back-end drivers in user-space
  - Similar to what FUSE and CUSE do for filesystems and character devices
- Uses io_uring for requests and responses to back-end
- Back-end drivers can use **ublksrv** library and daemon (RFP: #1051678)

Multi-generational LRU [6.1]

When RAM is nearly full, memory manager decides which pages of virtual memory to keep in RAM and which to reclaim (swap or flush):

- Theory: reclaim the Least Recently Used (LRU) pages
- Practice: it’s impossible to track exactly when pages are accessed, so use periodic scan and some heuristics
- Practice: sometimes LRU pages are needed again quite soon (thrashing)

MGLRU replaces the previous algorithms for determining LRU:

- Divides pages into 4 generations instead of just active/inactive
- Iterates over page tables rather than physical page frames
- Uses feedback loop (PID controller) to mitigate thrashing

MGLRU is enabled for most Debian architectures since Linux 6.4.

HID drivers in eBPF [6.3]

- HID (Human Interface Device) is a standard class for input devices on USB or Bluetooth — includes keyboards, mice, game controllers, etc.
- HID devices provide descriptors of their capabilities, so most can be handled by a single generic driver
- Some need special drivers to recognise custom keys, or to work around bugs in the device's descriptors
- The generic driver now supports doing those things, and many other kinds of filtering, with an eBPF program
  - Avoids the need to (re)build a custom driver for each kernel
  - Should avoid security bugs in descriptor parsing
  - **No infrastructure yet for collecting and distributing** such HID drivers
  - Not yet enabled in Debian

Rust for Linux [ongoing]

- Rust is a modern systems programming language designed to ensure memory-safety—preventing use-after-free, data races, etc.
- Many (most?) Linux kernel security vulnerabilities involve this sort of bug, so using Rust instead of C could improve security a lot
  - ...but replacing existing C code with Rust will be a long process
  - ...and currently this is “experimental”, so no core subsystem can use it yet
- Minimal support for Rust landed in 6.1; more APIs added later
- Currently no in-tree features written in Rust, but several out-of-tree drivers:
  - **asahi** — Apple GPU driver
  - **rnvme** — rewrite of NVMe block driver
  - **rust_binder** — rewrite of Binder IPC driver

In-kernel TLS [ongoing]

- Trusting the local network may have been reasonable in the ’80s, but is rather naïve today
- Most network filesystems and storage protocols are unencrypted and often unauthenticated, but this is now changing:
  - NFS over TLS now supported (server in 6.4, client in 6.5)
  - NVMe over TLS proposed; maybe available in 6.7
  - no patches for iSCSI over TLS yet
  - SMB has its own encryption and authentication
- TLS handshake and certificate validation are delegated to user-space:
  - **tlshd** daemon packaged in **ktls-utils**
  - Certificate validation still needs work; see upstream bug tracker

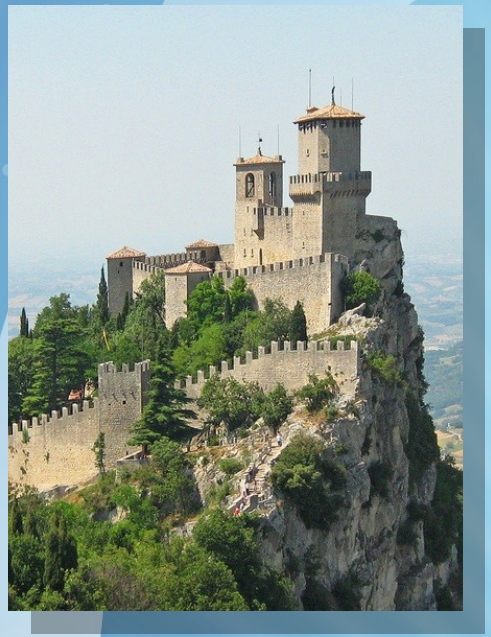
`cachestat` [6.5]

- New system call to query whether (part of) a file is cached in RAM
- Already possible with `mincore`, but by contrast `cachestat`:
  - Does not require the pages to be `mmapped`
  - Also exposes dirty and writeback states
  - Provides summary statistics instead of per-page flags
- Expected to be useful for:
  - Database engines such as PostgreSQL choosing whether to use an index
  - Applications that explicitly prefetch data, such as SQLite, to monitor how well this is working
  - Visibility of which files account for most memory usage

Netlink documentation [ongoing]

- Netlink is an extensible protocol used for configuring Linux networking and many other kernel subsystems
- Documentation of the protocol has been minimal:
  - It was supposed to replace socket `ioctl`-based APIs, but application developers often found those easier to understand
  - Even kernel networking developers introduced bugs in implementation that are now part of the protocol
- The kernel source (and linux-doc packages, and online docs) now include:
  - A Netlink Handbook for user-space developers working with netlink sockets
  - More minimal Netlink notes for kernel developers using its internal APIs

Security hardening



- Panic after multiple Oops or WARN events
  - Can protect against exploits that crash a lot
  - Limits configurable with sysctl
- New options for Control Flow Integrity (CFI):
  - [arm64] Boot-time choice of ROP protection: software shadow stacks or hardware PAC
  - [x86] FineIBT combines h/w Indirect Branch Tracking (IBT) and s/w type check
  - **Clang still required for software CFI**
- [s390x] Option to clear kernel stack on system call exit (STACKLEAK)

[x86] CPU bug mitigations

- Straight Line Speculation (SLS): speculation past unconditional branch or RET
  - Mitigation: compiler adds INT3 instructions
- Retbleed: train indirect branch predictor to mispredict return addresses
  - Mitigation: IBRS on Intel, return thunk on AMD
- Zenbleed: use-after-free in the vector register file — non-speculative!
  - Mitigation: microcode fix, set “chicken bit”, or disable AVX
- Gather Data Sampling (GDS): speculative access to stale vector register contents
  - Mitigation: microcode fix or disable AVX
- Speculative Return Stack Overflow (SRSO): train indirect branch predictor to train the return address predictor to mispredict RET instructions
  - Mitigation: it’s complicated

Packaging changes (1)

- Enabled support for various SoCs/platforms:
  - [arm64] Allwinner H6, Qualcomm SDA845; Renesas RZ/G2{L,M}; Rockchip RK{3328,3399,356x}
  - [armhf] NXP i.MX7
  - [riscv64] Allwinner D1, D1s; Microchip Polarfire; Renesas R9A07G043; StarFive JH7110
- Enabled support for Arm Coresight, many Intel CPU features, and CXL bus
- Changes to ABI “number” in kernel release string and package names:
  - Experimental uploads use **0** (instead of **rc***X* or **trunk**)
  - Backports use **0.deb***REL***.***ORIG* (distinguishing backports across multiple releases)
  - linux-kbuild packages now also incorporate the ABI “number”

Packaging changes (2)

- Enabled hardening options:
  - Kernel Electric Fence (KFENCE) partially mitigates buffer overflows and use-after-free; needs to be enabled at boot time
  - [arm64,powerpc,s390x,x86] Randomised kernel stack offset mitigates exploits that rely on uninitialised stack structures
- Disabled TIOCSTI – blocks privilege escalation through injecting input into privileged program sharing the terminal
- Fixes and improvements to the `test-patches` script:
  - Make all packages installable and coinstallable
  - Build faster: no `fakeroot`, no debug info
- [arm64,armhf] Enabled sound and speakup udebs for speech synthesis in the installer

Questions?

Credits & License

- Content by Ben Hutchings
  www.decadent.org.uk/ben/talks/
  License: GPL-2+
- Original OpenOffice.org template by Raphaël Hertzog
  raphaelhertzog.com/go/ooo-template
  License: GPL-2+
- Background based on “Serenity” theme by Edward Padilla
  wiki.debian.org/DebianArt/Themes/serenity
  License: GPL-2