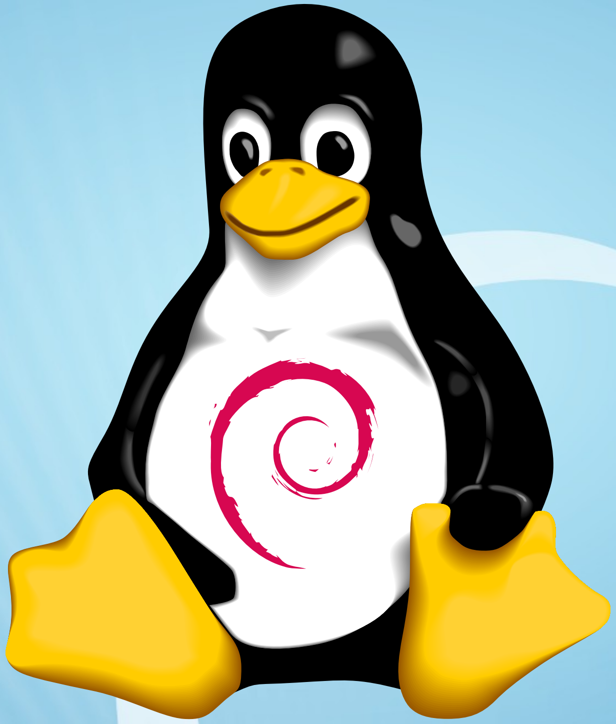
What's new in the Linux kernel

and what's missing in Debian

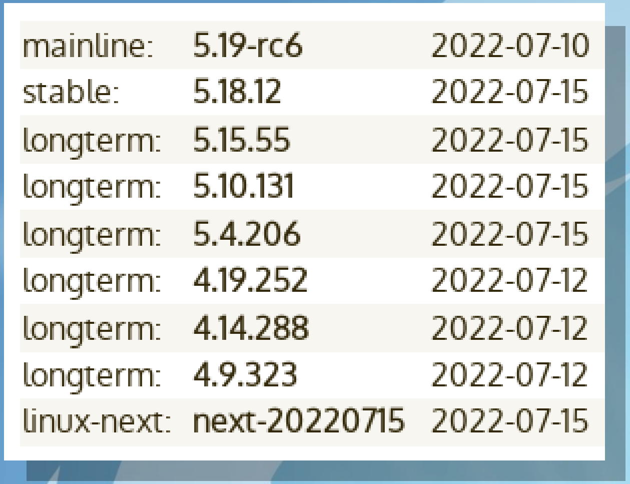


Ben Hutchings · DebConf 22

Ben Hutchings

- Working on Linux kernel and related code for Debian and in paid jobs for over 10 years
- Debian kernel and LTS team member, doing a lot of the kernel packaging and backporting work
- Formerly maintained Linux long-term stable branches needed by Debian

Linux releases early and often



mainline:	5.19-rc6	2022-07-10
stable:	5.18.12	2022-07-15
longterm:	5.15.55	2022-07-15
longterm:	5.10.131	2022-07-15
longterm:	5.4.206	2022-07-15
longterm:	4.19.252	2022-07-12
longterm:	4.14.288	2022-07-12
longterm:	4.9.323	2022-07-12
linux-next:	next-20220715	2022-07-15

- Linux has feature releases about 5 times a year, plus stable updates every week or two
- Some features aren't really ready in the 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 since Linux 5.10 (bullseye)

Recap of previous years' features (1)

RISC-V

Added support for:

- Virtualisation with KVM
- General performance monitoring events
- Tracing: {k,u}probes, ftrace
- kexec
- Transparent hugepages
- VMAP_STACK

Recap of previous years' features (2)

io_uring

- Added support for more operations
- Each process’s I/O executes in threads belonging to the process
  - Improved performance (no need to change context)
  - Reduces risk of using the wrong context
  - Made some more things work (e.g. `/proc/self` access)
- Integrated with the audit subsystem and LSMs
- More users in Debian: MariaDB, plocate, QEMU, Samba

Recap of previous years' features (3)

- Added features:
  - BTF in modules
  - Atomic operations
  - Timer callbacks
  - Bloom filters
  - CO-RE in kernel
  - Many new helper functions and hooks
- **Disabled** by default for users without CAP_SYS_ADMIN or CAP_BPF

seccomp bitmap optimisation [5.11]

- seccomp system call used to limit the system calls a task can use in future
- Used for sandboxing by systemd, bubblewrap, Docker, etc.
- Filters written in classic BPF, so flexible but slow
- Kernel now works out which system calls are always allowed and skips BPF execution for them
- Result: most sandboxed processes got faster

Landlock [5.13]

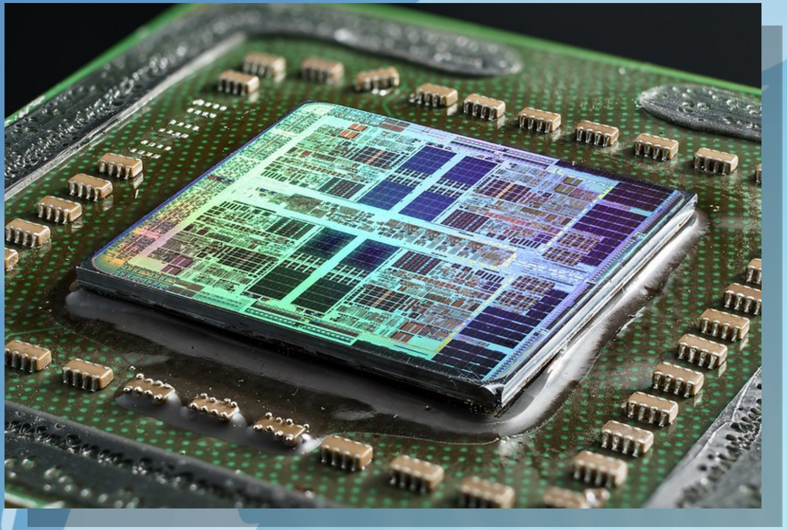
- A new Linux Security Module
- Inspired by FreeBSD’s Capsicum and OpenBSD’s `pledge`/`unveil` APIs
- Similar to `seccomp`, allows any process to restrict itself and its children
- Unlike `seccomp`, rules defined in terms of operations and paths
- Currently only controls filesystem operations

idmapped mounts [5.12]

- User namespaces remap uids and gids within a container, e.g. container uid 0 maps to global uid 1000
- Filesystems store global uids and gids
- Containers with different user namespaces could not share a rootfs, so container managers had to copy and chown files
- Solution: idmapped mounts, adding an additional mapping between global and on-disk ids
- Supported by most popular block-based filesystems, and overlayfs
- Used by systemd for “portable” services and home directories
- WIP to use this in containerd

PREEMPT_DYNAMIC [5.12]

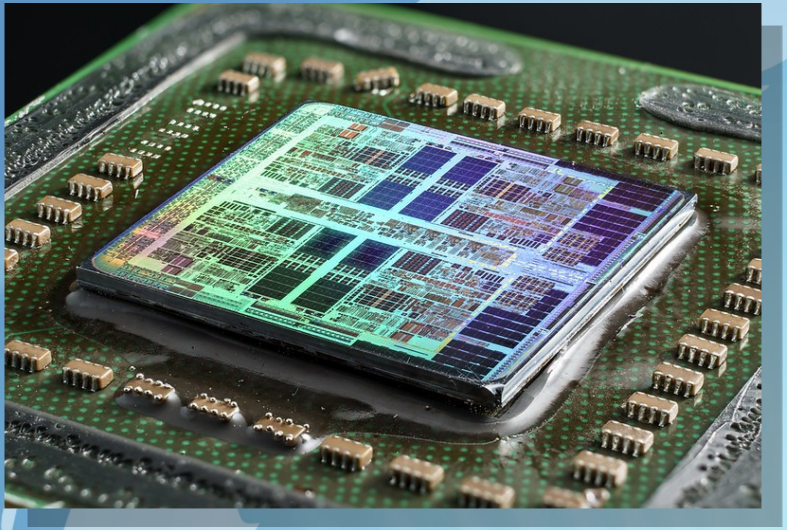
- Kernel config includes when to allow preemption in system calls—never, at specific points, or whenever it’s safe
- This is a trade-off between throughput and latency
- Desktops and servers typically want different preemption mode, and we compromise on the middle option
- Preemption mode can now be overridden with kernel parameter
  - Currently only on x86; could be enabled elsewhere
  - Not including RT preemption
- Should installer set the preemption mode e.g. based on whether you install a desktop?



Core scheduling [5.14] (1)

- SMT allows scheduling multiple concurrent threads on the same core, improving utilisation of CPU execution resources

- Resource sharing creates high bandwidth side-channels
  - Not a new problem, but speculative execution attacks have made it worse
- Resource sharing also causes unpredictable performance—bad for RT



Core scheduling [5.14] (2)

- Kernel *should not* let threads from different security contexts share a core
- Kernel *should not* schedule additional threads together with an RT thread

- Core scheduling allows user-space to define which threads can and can’t share a core
- containerd, lxc provide *options* to isolate containers with core scheduling
- QEMU/libvirt doesn’t yet seem to support isolating VMs

Real-Time Linux (PREEMPT_RT) [ongoing]

- PREEMPT_RT patch set adds the option for Linux to do real-time scheduling
- Useful for industrial and safety-critical applications, but also media production
- Developed since ~2005 by some upstream developers, but only small parts went upstream
- For 5.10: ~16,000 lines changed
- For 5.18: <4,000 lines changed, mostly small fixes for RT-incompatible driver code

Real-Time Linux tracing [5.14,5.17]

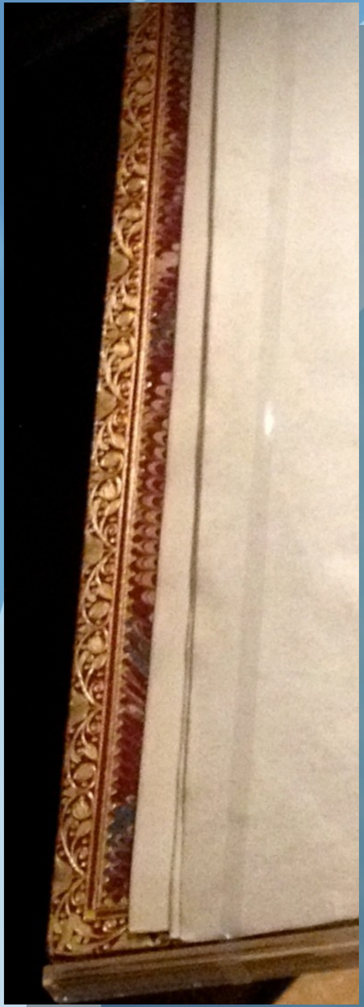
- osnoise and timerlat tracers added to support measurement of latency in real-time configurations
- `rtla` tool, included with kernel source, is a front-end to the tracers
- Will be shipped in a new binary package in the next 5.19 upload

ksmbd [5.15]

- New kernel-based SMB file server
- Higher performance than Samba, but without integration into Active Directory
- Managed with `ksmbd-tools`, already in Debian

Filesystem health reporting [5.16]

- `fanotify` has a new option to enable reporting of data corruption or I/O failures in the filesystem
- Needs filesystem support, currently limited to ext4
- User-space doesn’t seem to be using it yet—should UDisks or systemd be doing this?



Memory folios [5.16-ongoing]

- Kernel memory manager mostly deals with pages as defined by hardware MMU
- Pages can be grouped and managed as “compound pages”, but page and compound page pointers are same C type
- Kernel has lots of checks for whether a page is part of compound page, and bugs where wrong assumption was made
- Folio API introduces a distinct type for compound pages
- Should avoid this sort of bug and remove the need for a lot of run-time checks

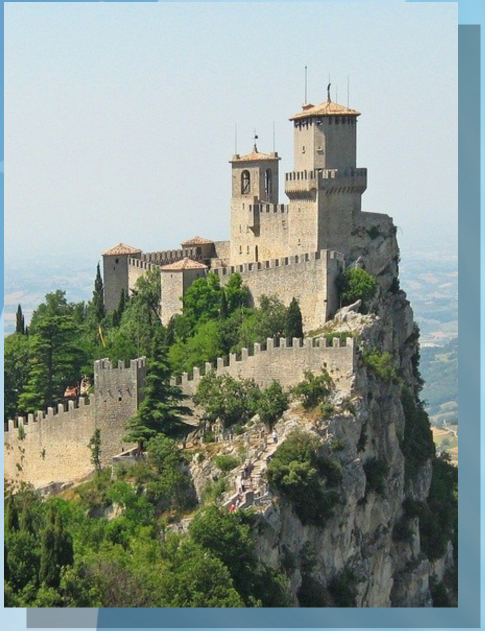
Write throttling rework [5.16]

- Block device writes are normally buffered in memory and written back later, but memory usage needs to be limited
- When a device can’t write data as fast as it’s being buffered (congestion), kernel makes writing tasks wait until congestion is cleared, or a timeout
- Not all drivers signalled congestion cleared, and block layer rewrite broke that completely, so tasks waited until timeout
- This is fixed in Linux 5.16, but it’s a complete reimplementation that won’t be backported

Random number generator

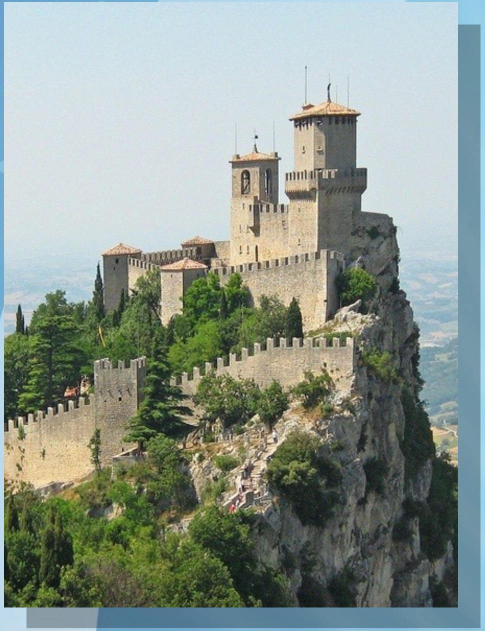
- Uses more conventional cryptography to combine entropy sources and to generate bits
- Should have higher performance, despite Intel RNG instructions getting slower
- Uses boot loader or UEFI as entropy source by default
- Uses CPU RNG as entropy source by default
- All above changes backported to stable!
- On most platforms, even `/dev/urandom` provides secure random bits immediately
- On arm64, uses hardware RNGs available through system firmware or special registers

Security hardening (1)



- [arm64,s390x,x86] Kernel stack randomisation mitigates attacks that involve reuse of stack buffers between system calls
    - Built-in by default but **needs a kernel parameter** to enable
- Stricter run-time bounds checking for `mem{cpy,move,set}` calls—overrunning array inside struct is now caught
- [armel,armhf,riscv64] VMAP_STACK prevents kernel stack overflow
    - Was already available and enabled on amd64, arm64, s390x

Security hardening (2)



- Control Flow Integrity (CFI) makes it harder to exploit bugs with ROP/JOP
  - [arm64] Software implementation; requires Clang
  - [x86] Limited hardware implementation (IBT); requires recent Intel CPU
  - **Neither enabled yet**

Packaging changes

- Added support for various SoCs/platforms:
  - [arm64] Microsoft Hyper-V; Qualcomm SDA845
  - [armhf] Marvell MMP{2,3}
  - [riscv64] Microchip Polarfire; StarFive JH7100
  - [x86] Intel Alder Lake, Emmitsburg, Jasper Lake, Lakefield
- MIPS configurations more consistent, and all MIPS architectures have a generic flavour
- linux-perf no longer matched to kernel version
- Implemented CI on Salsa:
  - Fixed all blhc warnings and lintian errors (but not all warnings)
  - Added a ‘quick’ build profile that should catch most regressions despite slow CI runners

Questions?

Credits & License (1)

- 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
- Hard drive image by Raimond Spekking
  commons.wikimedia.org/wiki/File:Toshiba_MK1403MAV_-_broken_glass_platter-93375.jpg
  License: CC BY-SA 4.0

Credits & License (2)

- Give Way sign by Roulex_45
  commons.wikimedia.org/wiki/File:Give-Way-sign.svg
  License: CC BY-SA 3.0
- Stopwatch image by Jerry
  www.flickr.com/photos/43437461@N00/4112797721
  License: CC BY 2.0
- Folio image by Jessie Chapman
  commons.wikimedia.org/wiki/File:William_Shakespeare%27s_first_folio.JPG
  CC BY-SA 4.0
- Traffic lights image by Old Photo Profile
  www.flickr.com/photos/10361931@N06/4747872021
  CC BY 2.0