Linux Plumbers Conference 2025

Report of Contributions

https://lpc.events/e/19

Contribution ID: **40** Type: **not specified**

Android MC

The Android Micro Conference brings the upstream community and Android systems developers together to discuss issues and changes to the Android platform and their dependencies and interactions with the Linux kernel, allowing for collaboration on solutions for upstream.

Some highlights of progress made since last year's MC:

- Community consolidation around the aosp-devs.org organization
- An example of device longevity discussions: Pixel devices in the field were up-reved to 6.1: https://www.androidauthority.com/pixel-phones-kernel-upgrade-march-update-3532360/
- Per-app v2 mem cgroups were shipped w/ Android 16, and vendors are starting to make use of it.
- For the 16k page size effort, discussions on CMA minimum alignments lead to relaxing this requirement for 16KB to reduce memory overhead from CMA regions. And discussions on ELF padding page-cache read-ahead and fault-around lead to later guard regions blocking fault-around and conclusion that the readahead issue may not be worth the effort to solve (for which android proceeded with only limiting fault-around similar to guard regions).
- Feedback and discussion from maintainers around ublk helped prioritize development as Ming provided some pointers for offloaded lz4 de-compression into the kernel with only the metadata I/Os being served from user-space. Future enhancements and investigation include suspend support for ublk process, BPF offloading and device mapper-style I/O control.
- The GBL presentation at LPC sparked collaboration with U-Boot maintainers - who began experimenting with GBL- and caught the attention of ARM SystemReady and EBBR leads, leading to ongoing discussions around standardizing Android boot.

Potential discussion topics for this year include:

- Updates and next-steps on Generic Boot Loader efforts
- Issues around supporting both 4KB and 16KB devices
- Deprecation of Ashmem and Rust Ashmem
- Efficient metric collection using BPF (iterators)
- Android Virtualization framework (AVF), updates, support, use cases
- AutoFDO for Android userspace and kernel
- Binder Priority Inheritance
- KUNIT Testing
- Discussions on collaborating with the new AOSP model

MC leads:
- Lukasz Luba lukasz.luba@arm.com

- Amit Pundir amit.pundir@linaro.org
- Mostafa Saleh smostafa@google.com
- Sumit Semwal sumit.semwal@linaro.org
- John Stultz jstultz@google.com
- Karim Yaghmour karim.yaghmour@opersys.com

Primary authors: PUNDIR, Amit; STULTZ, John (Google); YAGHMOUR, Karim (Opersys inc.); LUBA, Lukasz; SALEH, Mostafa; SEMWAL, Sumit (Linaro)

Presenters: STULTZ, John (Google); YAGHMOUR, Karim (Opersys inc.); SEMWAL, Sumit (Linaro)

Contribution ID: **48** Type: **not specified**

Build Systems MC

The Linux ecosystem supports a diverse set of methods for assembling complete, bootable systems, ranging from binary distributions to source-based systems, embedded platforms, and container-native environments. Despite differences in tooling and architecture, all of these systems face shared challenges: managing build complexity, ensuring security and reproducibility, maintaining cross-platform compatibility, and responding to increasing regulatory and supply chain scrutiny.

Building on the success of last year's microconference, we invite the community to continue the conversation with a broadened scope in 2025. This year, we aim to explore the intersection of build systems with CI/CD pipelines, supply chain security, critical infrastructure, and secure development practices. With legislation such as the Cyber Resilience Act, rising expectations for Software Bill of Materials (SBOMs), and mandates for reproducible and auditable builds, collaboration across the ecosystem has never been more essential.

This microconference provides a venue for architects, maintainers, and practitioners from all facets of the Linux build and distribution ecosystem to come together and share ideas, discuss pain points, and identify potential shared solutions.

Target communities and projects include (but are not limited to):

- General-purpose distributions: Debian, Fedora, Ubuntu, Arch Linux, openSUSE, Red Hat
- Source-based systems: Gentoo, NixOS, Guix, CRUX
- Embedded platforms: Yocto Project, OpenEmbedded, Buildroot, OpenWRT/LEDE, Android
- Container ecosystems: Docker, Podman, OCI, BuildKit, distrobuilders
- Immutable, image-based distributions: Flatcar, ParticleOS, Fedora Silverblue
- RTOS and hybrid build systems: Zephyr, RIOT, Mbed OS, FreeRTOS
- CI/CD and build orchestration: BuildStream, Buildbarn, Bazel, Jenkins, GitLab CI, GitHub Actions
- Compliance and supply chain security: SPDX, OSI, SBOM tooling, sigstore
- Broader open-source infrastructure efforts and standards bodies

Proposed discussion topics:

- Bootstrapping build systems and managing cross-compilation
- Integration of CI/CD pipelines into build workflows
- Securing the build lifecycle: from developer systems to package publication
- SBOM generation, license auditing, and legal/policy alignment
- Attestation, signing, and ensuring the software chain-of-trust
- Handling insecure or volatile upstream language-specific ecosystems (e.g., PyPI, npm, crates.io)
- Reproducible builds and deterministic output across toolchains
- Secure and scalable container build systems and image validation
- Immutable build pipelines for image-based systems and update strategies
- Resilience in build infrastructure for critical systems and edge deployments
- Patch sharing, lifecycle tracking, and cross-distro patch coordination

- Documentation, onboarding, and reducing the learning curve of complex build systems
- Long-term sustainability: mentoring, diversity, and community health of build toolchains

We welcome proposals beyond this list, particularly those that address emerging issues in the creation, validation, maintenance, and secure delivery of Linux-based software systems.

Improving coordination across build systems strengthens the foundations of the open-source ecosystem. Whether you're maintaining a distro, building firmware, managing containers, or designing infrastructure for high-assurance or real-time systems, this microconference is your forum to advance the state of Linux software construction and security.

Primary author: Mr WEBSTER, Behan (Linaro)

Co-author: BALISTER, Philip (OpenEmbedded)

Presenters: Mr WEBSTER, Behan (Linaro); Mr VASQUEZ, Frank; BALISTER, Philip (OpenEmbedded)

Contribution ID: **28** Type: **not specified**

Confidential Computing MC

The Confidential Computing microconferences of the past years have been a significant catalyst for better supporting trusted execution workloads in the Linux virtualization and general software stack. Since the last occurrence of the microconference AMD SEV-SNP and Intel TDX support for KVM were merged into the mainline Linux kernel as well as support for the Linux kernel running in ARM CCA environments.

But the open source software stack for confidential computing is still far from being complete. There remain many problems to be solved and functionality to enable. Some of the most important ongoing developments are:

-Support for large-page backing of confidential virtual machines (CVM).
- Privilege separation features in KVM via VM planes.
- Live migration of CVMs.
- Secure VM Service Module architecture and Linux support.
- Trusted I/O software architecture.
- Further topics to discuss are:
- Possible solutions for the full CVM (remote) attestation problem.
- Linux as a CVM operating system across hypervisors.
- Performance of CVMs.

The Confidential Computing microconference of 2025 wants to bring open source developers working on these topics together into productive discussions and to collaborate on solutions for the open problems.

Key attendees:
Ashish Kalra ashish.kalra@amd.com
Borislav Petkov bp@alien8.de
Dan Williams dan.j.williams@intel.com
Daniel P. Berrangé berrange@redhat.com
Dr. David Alan Gilbert dgilbert@redhat.com
David Hansen dhansen@linux.intel.com
David Kaplan David.Kaplan@amd.com
David Rientjes rientjes@google.com
Dhaval Giani dhaval.giani@amd.com
Dionna Amalie Glaze dionnaglaze@google.com
Elena Reshetova elena.reshetova@intel.com
James Bottomley James.Bottomley@HansenPartnership.com
Joerg Roedel joro@8bytes.org
Kirill A. Shutemov kirill.shutemov@linux.intel.com
Michael Roth michael.roth@amd.com
Mike Rapoport rppt@kernel.org
Paolo Bonzini pbonzini@redhat.com
Peter Fang peter.fang@intel.com
Peter Gonda pgonda@google.com
Sean Christopherson seanjc@google.com
Stefano Garzarella sgarzare@redhat.com
Tom Lendacky thomas.lendacky@amd.com

Primary authors: GIANI, Dhaval; ROEDEL, Joerg (AMD)

Presenters: GIANI, Dhaval; ROEDEL, Joerg (AMD)

Contribution ID: **25** Type: **not specified**

Containers and checkpoint/restore MC

The Containers and Checkpoint/Restore micro-conference focuses on both userspace and kernel related work.

The micro-conference targets the wider container ecosystem ideally with participants from all major container runtimes as well as init system developers.

The microconference will be discussing recent advancements in container technologies with some of the usual candidates being:

- VFS API improvements (new system calls, idmap, ⋯)
- CGroupV2 feature parity with CGroupV1 and migration path
- Dealing with the eBPF-ification of the world
- Mediating and intercepting complex system calls
- Making user namespaces more accessible
- Verifying the integrity of containers
- Improving the set of resource limits available

On the checkpoint/restore front, some of the potential topics include:

- Making CRIU work with modern Linux distributions
- Handling GPUs
- Restoring FUSE daemons
- Dealing with restartable sequences
- Use of eBPF
- Support of new kernel features
- Supporting shadow stack (x86, arm64)
- Support for madvise(MADV_GUARD_INSTALL)
- Support for mseal()
- Support for pidfd C/R, including process exit information

And quite likely a variety of other container and checkpoint/restore topics as things evolve between now and the event.

Past editions of this micro-conference have been the source of many developments in the Linux kernel, including:

- PIDfds
- VFS idmap (and adding it to a slew of filesystems)
- FUSE in user namespaces
- Unprivileged overlayfs
- Time namespace

- A variety of CRIU features and checkpoint/restore kernel interfaces with the latest among them being
- Unpriviledged checkpoint/restore
- Support of rseq(2) checkpointing
- IMA/TPM attestation work

Primary authors: REBER, Adrian (Red Hat); Mr BRAUNER, Christian; RAPOPORT, Mike; GRABER, Stéphane (Zabbly)

Presenters: REBER, Adrian (Red Hat); Mr BRAUNER, Christian; RAPOPORT, Mike; GRABER, Stéphane (Zabbly)

Contribution ID: **43** Type: **not specified**

Embedded & Internet of Things MC

The Embedded and IoT Micro-conference is a forum for developers to discuss all things Embedded and IoT. Topics include tools, telemetry, device drivers, protocols and standards in not only the Linux kernel but also Real-Time Operating Systems.

Current Problems that require attention (stakeholders):

- Boot time optimizations (Tim Bird, Khasim Syed Mohammed)
- Kernel size and memory requirement optimizations
- CAN subsystem (Marc Kleine-Budde, Oleksij Rempel)
- Linux-wpan subsystem (Stefan Schmidt, Miquel Raynal, Alexander Aring)
- Sync device tree description of hardware between U-Boot, Linux and Zephyr (Nishanth Menon)
- Generic MCU communication driver (Schuyler Patton)
- Any other topics embedded and IoT

We hope you will join us either in-person or remote for what is shaping up to be another great event full of collaboration, discussion, and interesting perspectives.

Primary author: SCHMIDT, Stefan

Co-author: LÜBBE, Jan (Pengutronix)

Presenters: LÜBBE, Jan (Pengutronix); SCHMIDT, Stefan

Contribution ID: **4** Type: **not specified**

Gaming on Linux MC

The Gaming on Linux Microconference welcomes the community to discuss a broad range of topics around performance improvements for Gaming devices running Linux. Gaming on Linux has pushed the kernel to improve in several areas and has helped create new features for Linux, such as the futex_waitv() syscall, the Unicode subsystem, HDR support, and much more. Although they were initially created for gaming use cases, now they are used in different scenarios.

The potential topics for this year are around a lot of subsystems in the kernel, including:

- virtualization of other OSs and emulation
- cgroups for 3D resources
- schedulers focused in gaming workloads
- optimization of locking mechanisms
- filesystems
- power management optimizations
- debug data collection
- memory management challenges in gaming scenarios

Primary authors: ALMEIDA, André (Igalia); VERNET, David (Meta)

Presenters: ALMEIDA, André (Igalia); VERNET, David (Meta)

Contribution ID: **34** Type: **not specified**

Kernel Memory Management MC

Memory management keeps on being exciting. With a lot of activity on all different kinds of projects, some more controversial subjects that might be worth discussing this year:

- Making Transparent Huge Pages more … transparent (toggles, policies, khugepaged, …)
- Making (m)THP/large folios a first-class citizen in MM
- What other improvements might we see from mTHP?
- Where to use eBPF in MM, and where not
- Ongoing challenges with memdescs (e.g., allocation/freeing/walking)
- How might we make allocations guaranteed to not fail?
- Which CXL use cases do we want to support, and how far should we go?
- Challenges with hypervisor live-update, and the integration into other subsystems (MM, drivers, etc)
- guest_memfd and the interaction with other MM subsystems (hugetlb, GUP, …)
- Making hugetlb less weird

Primary authors: HILDENBRAND, David (Red Hat); WILCOX, Matthew (Oracle)

Presenter: HILDENBRAND, David (Red Hat)

Contribution ID: **1** Type: **not specified**

Kernel Testing & Dependability MC

The Kernel Testing & Dependability Micro-Conference (a.k.a. Testing MC) focuses on advancing the current state of testing of the Linux Kernel and its related infrastructure.

Building upon the momentum from previous years, the Testing MC's main purpose is to promote collaboration between all communities and individuals involved with kernel testing and dependability. We aim to create connections between folks working on related projects in the wider ecosystem and foster their development. This should serve applications and products that require predictability and trust in the kernel.

We ask that all discussions focus on some identified issues, aiming at finding potential solutions or alternatives to resolving them. The Testing MC is open to all topics related to testing on Linux, not necessarily in the kernel space.

In particular, here are some popular topics from past editions:

- KernelCI: Maestro, kci-dev, kci-deploy, kci-gitlab, new dashboard, KCIDB
- Improve sanitizers: KFENCE, KCSAN, KASAN, UBSAN
- Using Clang for better testing coverage: Now that the kernel fully supports building with Clang, how can all that work be leveraged into using Clang's features?
- Consolidating toolchains: reference collection for increased reproducibility and quality control.
- How to spread KUnit throughout the kernel?
- Building and testing in-kernel Rust code.
- Identify missing features that will provide assurance in safety critical systems.
- Which test coverage infrastructures are most effective to provide evidence for kernel quality assurance? How should it be measured?
- Explore ways to improve testing framework and tests in the kernel with a specific goal to increase traceability and code coverage.
- Regression Testing for safety: Prioritize configurations and tests critical and important for quality and dependability.
- Transitioning to test-driven kernel release cycles for mainline and stable: How to start relying on passing tests before releasing a new tag?
- Explore how do SBOMs figure into dependability?
- kernel benchmarking and kernel performance evaluation

Things accomplished from last year:

- progress on Rust testing
- kci-dev is currently used in production for interacting with KernelCI results
- Follow up discussions on kselftest mailing list about "Adding benchmarks results support to KTAP/kselftest"
- Proposal of kci-gitlab sent to kselftest mailing list

Primary authors: TACHIBANA, Arisu; TUCKER, Guillaume; LEVIN, Sasha; KHAN, Shuah (The Linux Foundation)

Presenter: TACHIBANA, Arisu

Contribution ID: **9** Type: **not specified**

Live Update MC

Live Update is a specialized reboot process where selected devices are kept operational and kernel state is preserved and recreated across a kexec. For devices, DMA and interrupts may continue during the reboot.

The primary use-case of Live Update is to enable hypervisor updates in cloud environments with minimal disruption to running virtual machines. During a Live Update, a VM can pause and its state is stored to memory while the hypervisor reboots. PCIe devices attached to those VMs (such as GPUs, NICs, and SSDs), are kept running during the Live Update. After the reboot, VMs are recreated and restored from memory, reattached to devices, and resumed. The disruption is limited to the time it takes to complete this entire process.

With Live Update infrastructure in place, other use-cases may emerge, like for example preserving the state of GPU doing LLM, freezing running containers with CRIU, and preserving large in-memory databases.

The Live Update and state persistence functionality touch on different parts of the kernel and this microconference aims to bring together people from different subsystems. Upstream support for Live Updates is still in its infancy and there are a lot of unsolved aspects that will benefit from direct communication.

Key problems that will be discussed:

Support for memfd/guest_memfd/hugetlb/tmpfs
Preserving the state of VFIO, IOMMUFD, and IOMMU drivers.
Kernel <-> userspace interaction during Live Update
Integration of Live Update with PCI and Device Model
Persistence of movable memory
Leveraging suspend/resume functionality for device state preservation
Optimizing kernel shutdown and boot times
Automated Testing of Live Updates

Key attendees:

Pasha Tatashin
David Matlack
David Rientjes
Chris Li
Bjorn Helgaas
Samiullah Khawaja
Vipin Sharma
Josh Hilke
Changyuan Lyu
Alex Graf
David Woodhouse
James Gowans
Pratyush Yadav
Jason Gunthorpe
Mike Rapoport
Alex Williamson

Primary authors: GRAF, Alexander; MATLACK, David (Google); RAPOPORT, Mike; TATASHIN, Pasha

Presenters: GRAF, Alexander; MATLACK, David (Google); RAPOPORT, Mike; TATASHIN, Pasha

Contribution ID: **46** Type: **not specified**

Power and Thermal management MC

The Power Management and Thermal Control microconference is about all things related to saving energy and managing heat. Among other things, we care about thermal control infrastructure, CPU, and device power-management mechanisms, energy models, and power capping.

This year has been mainly focused on the maintenance of the frameworks, resulting in cleanups setting up the scene for more improvements and connecting the user space to the kernel:

- Thermal thresholds have been finished: https://patch.msgid.link/20240923100005.2532430-2-daniel.lezcano@linaro.org
- Thermal library and thermal engine skeleton taking benefit of thresholds: https://patch.msgid.link/20241022155147.46
  5-daniel.lezcano@linaro.org
- Performance QoS RFC has been posted: https://lore.kernel.org/all/20250505161928.475030-1-daniel.lezcano@linaro.org/

Embedded systems with very high performances and a higher integration, automotive systems with lifespan and reactivity constraints, push the PM frameworks to their limits.

Big topics to be addressed:

- A performance QoS has been sent as an RFC. It apparently needs more discussion as the approach seems to not satisfy all the parties. This framework is needed to allow the user space to interact with the performance on different devices via a unified interface and coexist with the kernel decisions.
- The generalization of the telemetry on embedded systems to capture the energy, the power, and the temperature at high rate. How can the kernel use these data and for what ?
- In the context of electric vehicles, power management is important to insure the largest life span of the hardware. Some operation modes, like the sentry mode, challenge several bricks of the power management like resume from suspend time.
- The energy model can no longer be static in the kernel and must be dynamically adjusted. Depending on the running scenario, there can be a significant gap between the computed and real power consumption, leading to inappropriate kernel decisions.
- Embedded systems are looking for a multi suspend states mechanism like a system-wide C-state. It requires more discussion, which were already initiated at the last LPC.
- The sensor aggregation did not reach a consensus because there is a different perception on the goal of the aggregation and its usage, resulting in a different implementation

---

Key attendees:

- Rafael J. Wysocki
- Lukasz Luba
- Vincent Guittot

- Saravana Kannan
- Sriniva Pandruvada
- Ulf Hansson
- Deepti Jaggi
- Prasad Sodagudi
- Manaf Meethalavalappu Pallikunhi
- Tudor Ambarus
- Amit Kucheria
- Srinivas Kandagatla

Primary author: LEZCANO, Daniel (Linaro)

Co-author: WYSOCKI, Rafael (Intel Open Source Technology Center)

Presenter: LEZCANO, Daniel (Linaro)

Contribution ID: **29** Type: **not specified**

RISC-V MC

We'd like to propose bringing back the RISC-V Microconference at Linux Plumbers 2025. As the RISC-V ecosystem continues to grow, so does the importance of having a space where developers, hardware vendors, toolchain maintainers, and distro folks can come together to solve real-world problems. This microconference has always been a great venue for open, technical discussions that help move RISC-V Linux support forward —from core architecture work to platform enablement and userspace coordination. In general, anything that touches both Linux and RISC-V is fair game, but the conversation often centers around a few common themes we have been following:
Supporting new RISC-V ISA features in Linux, especially vendor-specific or upcoming standardized extensions
Enabling RISC-V-based SoCs, which often involves working across various Linux subsystems in addition to the core arch/riscv code
Coordinating with distributions and toolchains to ensure consistent and correct userspace-visible behavior
Possible Topics
It's still early to lock down a full agenda, but a number of topics have been circulating on the various mailing lists lately are:
The issues with WARL discovery
Why CFI is not merged yet (if not merged by then)
ACPI enablement progress and remaining gaps (e.g ACPI WDAT watchdog)
How to handle RVA profiles in Linux kernel ?
QoS

Key Stakeholders
Sorry if I missed anyone, but here's a quick list of folks who've regularly shown up and helped keep the conversation moving at past RISC-V MC:
RISC-V contributors: Palmer, Atish, Anup, Conor, Sunil, Charlie, Bjorn, Alex, Clément, Andrew, Deepak —and probably a few more I'm forgetting.

SoC folks: Arnd, Conor, Heiko, Emil, Drew —with more SoC families bringing up RISC-V support, this group keeps expanding.

We've also consistently had great input from contributors across other architectures like arm, arm64, ppc, mips, and loongarch. Since we all end up touching shared code (especially in areas like drivers or platform support), these cross-arch discussions have been super valuable —and we expect even more of that this year as shared SoC platforms become more common.
Accomplishments post 2024 Microconference
Ftrace improvements 1
System MSI support [2]
RIMT patches available on lore[3]
CFI series is at v17 4 https://lore.kernel.org/linux-riscv/174890236299.925497.5731685320676689356.git-patchwork-notify@kernel.org/#r
[2] https://lore.kernel.org/linux-riscv/20250611062238.636753-13-apatel@ventanamicro.com/
[3] https://lore.kernel.org/linux-riscv/20250610104641.700940-1-sunilvl@ventanamicro.com/
[4] https://lore.kernel.org/linux-riscv/20250604-v5_user_cfi_series-v17-0-4565c2cf869f@rivosinc.com/#r

Primary authors: PATRA, ATISH (Rivos); TÖPEL, Björn (N/A); DABBELT, Palmer (Google)

Presenters: PATRA, ATISH (Rivos); TÖPEL, Björn (N/A); DABBELT, Palmer (Google)

Contribution ID: **47** Type: **not specified**

Rust MC

Rust is a systems programming language that is making great strides in becoming the next big one in the domain. Rust for Linux is the project adding support for the Rust language to the Linux kernel.

Rust has a key property that makes it very interesting as the second language in the kernel: it guarantees no undefined behavior takes place (as long as unsafe code is sound). This includes no use-after-free mistakes, no double frees, no data races, etc. It also provides other important benefits, such as improved error handling, stricter typing, sum types, pattern matching, privacy, closures, generics, etc.

This microconference intends to cover talks and discussions on both Rust for Linux as well as other non-kernel Rust topics.

Possible Rust for Linux topics:

- Rust in the kernel: status updates and discussion on next steps.
- Use cases for Rust around the kernel: subsystems, drivers, other modules…
- Developing-related discussions: how to abstract existing subsystems safely and API design, coding guidelines, safety guidelines…
- Upstreaming process: guidance on how to get into mainline, strategies that have worked for Rust code in the past, getting involved…
- Maintanance: the new subentries and branches, the proposed cross-subsystem subteams (e.g. the safety team), scaling work for the future, any cross-subsystem issues…
- Infrastructure: build system, documentation, testing and CIs, maintenance, unstable features, architecture support, stable/LTS releases, Rust versioning, third-party crates…
- klint.
- pinned-init.
- The future of GCC builds.

Possible Rust topics:

- Language and standard library: discussion on upcoming features, stabilization of the remaining features the kernel needs, memory model, the 2024 edition…
- Compilers and codegen: rustc improvements, LLVM and Rust, rustc_codegen_gcc, gccrs…
- Other tooling and new ideas: Coccinelle for Rust, bindgen, Compiler Explorer, Cargo, Clippy, Miri…
- Educational material.
- Any other Rust topic within the Linux ecosystem.

Last year was the third edition of the Rust MC and the focus was on discussing the ongoing efforts by different parties that are upstreaming new Rust abstractions and drivers (Giving Rust a chance for in-kernel codecs, hrtimer Rust Abstractions, Atomics and memory model for Rust code in kernel). We also had a topic related to improving the ergonomics and tooling around Rust in the kernel (Coccinelle for Rust) and a tutorial session to help others learn Rust (Introduction to Rust: Quality of Life Beyond Memory Safety), as well as a "Birds of a Feather" slot for open discussion on other topics.

Since the MC last year, it is easy to notice how Rust work around the kernel is accelerating: patches and contributors keep growing, new MAINTAINERS entries have been created and new maintainers have stepped up, new use cases have been developed and/or merged (e.g. the AMCC QT2025 PHY driver and Android ashmem), new projects have been announced (e.g. Tyr)… Even some end users of Linux distributions have already interacted with Rust kernel code in the wild (via the QR code panic screen). This all signifies success, but also poses new challenges ahead.

Note that this year submissions for the Rust MC should aim to be more discussion oriented.

Suggested attendees: the Rust for Linux team (Miguel Ojeda, Boqun Feng, Gary Guo, Benno Lossin, Andreas Hindborg, Alice Ryhl, Trevor Gross, Danilo Krummrich), Abdiel Janulgue, Alexandre Courbot, Alistair Francis, Arnaldo Carvalho de Melo, Bjorn Helgaas, Burak Emir, Christian Brauner, Christian Schrefl, Daniel Almeida, Dave Airlie, David Gow, Dirk Behme, Fiona Behrens, Frederic Weisbecker, FUJITA Tomonori, Greg Kroah-Hartman, Igor Korotin, Ingo Molnar, Jocelyn Falempe, Joel Fernandes, Kees Cook, Liam R. Howlett, Lorenzo Stoakes, Luis Chamberlain, Lyude Paul, Masahiro Yamada, Matthew Maurer, Nathan Chancellor, Paolo Bonzini, Paul E. McKenney, Peter Zijlstra, Remo Senekowitsch, Rob Herring, Robin Murphy, Sami Tolvanen, Stephen Boyd, Tamir Duberstein, Tejun Heo, Thomas Gleixner, Viresh Kumar, Will Deacon, Yury Norov…

Primary author: OJEDA, Miguel

Presenter: OJEDA, Miguel

Contribution ID: **2** Type: **not specified**

Safe Systems with Linux MC

Description

As Linux continues to be deployed in systems with varying criticality constraints, progress needs to be made in establishing consistent linkage between code, tests, and requirements, to improve overall efficiency and ability to support necessary analysis.
This MC addresses critical challenges in expectation management (aka requirements tracking), documentation, testing, and artifact sharing within the Linux kernel ecosystem. While tests are contributed for the code, traditionally the underlying requirement that the tests satisfies is likewise not documented in a structured manner. This has resulted in a large amount of "tribal knowledge" associated with subsystems, which results in technical debt when maintainers stop working on subsystems.

Taking in the feedback from last year's "Safe Systems with Linux" miniconference 1, on how we can improve the documentation of the kernel's design [1a] the ELISA (Enabling Linux in Safety Applications) community has focused on prototyping a template for capturing the requirements with volunteer linux kernel subsystem maintainers. The ELISA architecture team 2 has been meeting weekly and has developed a structured approach for documenting testable expectations with a template that allows embedding requirements directly with relevant code (as requested in the initial workshop) while maintaining machine readability and forming a base for improving testing with initiatives like KernelCI. The prototype format got initial review and feedback in December at the ELISA workshop at Goddard [3] and after incorporating that feedback in the workshop in Lund in May [4].

Initial pilots in the TRACING subsystem [5] have demonstrated the value of this approach, even resulting in the identification and fixing of previously unknown issues. [6,7]

Building on the last year's discussions, the goal of this miniconference is to get wider feedback from additional maintainers and developers of different subsystems on the approach being proposed.

Potential Topics

- **Progress on Linux Kernel Requirements Framework**
  Discussing the SPDX-based template for low-level requirements, lessons learned from initial pilots, and plans for wider adoption.
- **Technical Debt Reduction**
  How documented requirements capture understanding of original functionality, and can be leveraged for verification when code needs to be rewritten (ie. C to Rust), etc.
- **Requirements-Driven Testing**
  How documented requirements can drive test case development and validation. Connecting relevant test cases with specific requirements and code, should be able to yield more efficient testing.
- **Semantic Aspects of Kernel Requirements**
  Exploring how to properly document expected behaviors with consideration for design elements that impact or are impacted by these behaviors.
- **Practical Implementation Challenges**
  Addressing the balance between detailed requirements documentation and maintaining kernel development velocity.

- **Required tools for automation**
  Progress on tools to generate, validate, and track work products increasing dependability throughout the kernel development process.
- **Industry Adoption**
  How safety-critical industries are beginning to leverage these developments for certification and compliance purposes. How their safety engineers can participate in contributing formalized requirements to the kernel and providing linkage.
- **Requirements as an Education Tool**
  How linux kernel documentation can mine the requirements, and help new contributors understand kernel functionality and design intent and attract new upsteam developers

Summary

Last year, we established the need for better documentation of requirements in safe systems. This year, we will showcase concrete progress, including a working framework for Linux Kernel Low Level Requirements with practical implementations in the TRACING subsystem. This MC aims to bring together kernel maintainers, developers, with safety architects and industry stakeholders to expand adoption of these practices and address remaining challenges in building safe systems with Linux. It should engage with testing and documentation centric activities and how all parts can link together.

Potential Participants

Steve Rostedt
Greg Kroah-Hartman
Thomas Gleixner
Jonathan Corbet
Tim Bird
Shuah Khan
Gustavo Padovan
Gabrielle Paoloni
Chuck Wolber
Luigi Pellecchia
Alessandro Carminati
Wolfram Sang (Renesas BSP)
Vincent Mailhol (CAN Subsystem)
Kate Stewart
Philipp Ahmann
Nicole Pappler

References

\1 LPC 2024 Safe Systems with Linux Miniconf: https://lpc.events/event/18/sessions/187/#20240920

[1a] Kernel design documentation improvement: https://www.youtube.com/watch?v=stqGiy85s_Y

\2 ELISA Architecture meetings: https://lists.elisa.tech/g/safety-architecture

[3] NASA workshop: https://www.youtube.com/watch?v=_N3l_EEV8uM

[4] Link to Lund workshop session: https://drive.google.com/file/d/1−e82k80_D79ycJdFbEwLQ0kpbD0pMR9/view?usp=sha

[5] Drafting requirements in tracing subsystem: https://github.com/torvalds/linux/compare/master…elisa-tech:linux:linux_requirements_wip

[6] Patch on LKML: https://lkml.org/lkml/2025/3/21/1128

[7] Patch on LKML: https://lkml.org/lkml/2025/5/21/850

Primary authors: STEWART, Kate (Linux Foundation); AHMANN, Philipp (Etas GmbH (BOSCH))

Presenters: STEWART, Kate (Linux Foundation); AHMANN, Philipp (Etas GmbH (BOSCH))

Contribution ID: **17** Type: **not specified**

sched_ext: The BPF extensible scheduler class MC

sched_ext1 is a Linux kernel feature which enables implementing safe task schedulers in BPF, and dynamically loading them at runtime. sched_ext enables safe and rapid iterations of scheduler implementations, thus radically widening the scope of scheduling strategies that can be experimented with and deployed, even in massive and complex production environments.

This MC is the space for the community to discuss the developments of sched_ext, its impact on the community, and to outline future strategies aimed at improving the integration with the other Linux kernel subsystems.

Last year the sched_ext MC proved highly productive in facilitating coordination with distribution maintainers, allowing us to clarify their requirements and ease potential maintenance burdens. This collaboration directly contributed to upstream changes, including patches such as 2.

Ideas of topics to be discussed include (but are not limited to):
- Use of BPF arenas for task/CPU context sharing between kernel, BPF, and user space
- Composable schedulers/scheduler libraries with BPF arenas
- Deadline server(s) for the SCHED_EXT class
- Integration with other scheduling-related features (RCUs, proxy execution, PREEMPT_RT, etc.)
- Potential integration with other Linux subsystems (e.g., Rust-for-Linux)
- Scheduling for gaming and latency-sensitive workloads
- User-space scheduling
- Tickless scheduling
- Tools and benchmarks to analyze and understand scheduler activities

While we already have a tentative schedule with existing talk proposals to cover the topics mentioned above, we are also planning to open a public CFP to accept additional topics to discuss. Time permitting, we are open to readjust the schedule to accommodate further discussions that are relevant to the Linux community.

1 https://github.com/sched-ext/scx

2 https://lore.kernel.org/all/20240921193921.75594-1-andrea.righi@linux.dev/

Primary author: RIGHI, Andrea (NVIDIA)

Co-authors: Mr FERNANDES, Joel (NVIDIA); Mr HODGES, Daniel (Meta); MIN, Changwoo (Igalia)

Presenters: Mr FERNANDES, Joel (NVIDIA); Mr HODGES, Daniel (Meta); MIN, Changwoo (Igalia); RIGHI, Andrea (NVIDIA)

Contribution ID: **23** Type: **not specified**

Scheduler and Real-Time MC

For Linux Plumber 2025, we propose a joint microconference for Real Time and Scheduler as in the past. These two areas have always been tightly linked and continue to generate cross functional changes especially after PREEMPT_RT has been merged. The scheduler is at the core of Linux performance; With different topologies and workloads, giving the user the best experience possible is challenging, from low latency to high throughput and from small power-constrained devices to HPC.

Since last year's micro conference, progress has been made on the following topics:
- Progress on proxy execution
- https://lore.kernel.org/all/20241011232525.2513424-1-jstultz@google.com/
- https://lore.kernel.org/all/20250602221004.3837674-1-jstultz@google.com/
- Defer throttle when task exits to user
- https://lore.kernel.org/all/20250520104110.3673059-1-ziqianlu@bytedance.com/
- The EEVDF scheduler responsiveness
- https://lore.kernel.org/all/20250418151225.3006867-1-vincent.guittot@linaro.org/
- https://lore.kernel.org/all/20250209235204.110989-1-qyousef@layalina.io/

Some topics also continued to be discussed at the OSPM conference: http://retis.sssup.it/ospm-summit/

Ideas of topics to be discussed include (but are not limited to):
- Improve responsiveness of fair tasks
- Improvements on EEVDF
- Adding more usages of push callback
- Improve PREEMPT_RT
- Defer throttle to user space
- IPI impact
- Improve Locking and priority inversion
- Proxy execution
- Impact of new topology, including hybrid or heterogeneous system
-Taking into account task profile
- Improvements on SCHED_DEADLINE and DL server
- Tooling for debugging low latency analysis

This is not an exhaustive list. We welcome all proposals related to process scheduling. The goal is to discuss open problems, preferably with patch set submissions already being discussed on the mailing list. Presentations are meant to be limited to 2 or 3 slides intended to seed a discussion and debate - allowing for high bandwidth discussion with key stakeholders in the same room.

Key attendees:
- Ingo Molnar
- Peter Zijlstra
- Juri Lelli
- Vincent Guittot
- Dietmar Eggemann
- Steven Rostedt
- Ben Segall
- Mel Gorman
- Valentin Schneider
- Thomas Gleixner
- John Stulz

- Sebastian Andrzej Siewior
- K Prateek Nayak
- Shrikanth Hegde
- Phil Auld
- Dhaval Giani
- Clark Williams

Primary authors: DHAVAL, Giani; LELLI, Juri (Red Hat); AULD, Phil (Red Hat); ROSTEDT, Steven; GUITTOT, Vincent (Linaro)

Presenter: GUITTOT, Vincent (Linaro)

Contribution ID: **38** Type: **not specified**

System Boot and Security MC

The System Boot and Security Microconference has been a critical platform for enthusiasts and professionals working on firmware, bootloaders, system boot, and security. This year, once again, we want to focus on the challenges that arise when upstreaming boot process improvements to the Linux kernel and bootloaders. Our experience shows that the introduction of new and/or not well-known technologies into the kernel are especially difficult. The TrenchBoot project is a very good example here, but we think others are also impacted. So, it would be good to take all project stakeholders in one room and think what does not work, what can be improved, etc. Though we are also happy to hear and discuss what is currently happening in other areas related to platform initialization and OS boot. Especially discussion about obstacles, not only technical ones, during upstreaming and finding solutions during the MC can be very valuable for various projects and the audience.

We welcome talks on the following things that can help achieve the goals mentioned above:

- TrenchBoot, tboot,
- TPMs, HSMs, secure elements,
- Roots of Trust: SRTM and DRTM,
- Intel TXT, SGX, TDX,
- AMD SKINIT, SEV,
- ARM DRTM,
- Growing Attestation ecosystem,
- IMA,
- TianoCore EDK II (UEFI), SeaBIOS, coreboot, U-Boot, LinuxBoot, hostboot,
- Measured Boot, Verified Boot, UEFI Secure Boot, UEFI Secure Boot Advanced Targeting (SBAT),
- shim,
- boot loaders: GRUB, systemd-boot/sd-boot, network boot, PXE, iPXE,
- UKI,
- u-root,
- OpenBMC, u-bmc,
- legal, organizational, and other similar issues relevant to people interested in the system boot and security.

Primary authors: KIPER, Daniel; KRÓL, Piotr (3mdeb)

Presenters: KIPER, Daniel; KRÓL, Piotr (3mdeb)

Contribution ID: **35** Type: **not specified**

VFIO/IOMMU/PCI MC

The PCI interconnect specification, the devices that implement it, and the system IOMMUs that provide memory and access control to them are nowadays a de-facto standard for connecting high-speed components, incorporating more and more features such as:

```
- Address Translation Service (ATS)/Page Request Interface (PRI)
- Single-root I/O Virtualization (SR-IOV)/Process Address Space ID (PASID)
- Shared Virtual Addressing (SVA)
- Remote Direct Memory Access (RDMA)
- Peer-to-Peer DMA (P2PDMA)
- Cache Coherent Interconnect for Accelerators (CCIX)
- Compute Express Link (CXL)/Data Object Exchange (DOE)
- Component Measurement and Authentication (CMA)
- Integrity and Data Encryption (IDE)
- Security Protocol and Data Model (SPDM)
```

These features are aimed at high-performance systems, server and desktop computing, embedded and SoC platforms, virtualisation, and ubiquitous IoT devices.

The kernel code that enables these new system features focuses on coordination between the PCI devices, the IOMMUs they are connected to, and the VFIO layer used to manage them (for userspace access and device passthrough) with related kernel interfaces and userspace APIs to be designed in-sync and in a clean way for all three sub-systems.

The VFIO/IOMMU/PCI MC focuses on the kernel code that enables these new system features, often requiring coordination between the VFIO, IOMMU and PCI subsystems.

Following the success of LPC 2017, 2019, 2020, 2021, 2022, 2023 and 2024 VFIO/IOMMU/PCI MC, the Linux Plumbers Conference 2024 VFIO/IOMMU/PCI track will focus on promoting discussions on the PCI core and current kernel patches aimed at VFIO/IOMMU/PCI subsystems. Specific sessions will focus on discussions that require coordination between the three subsystems.

See the following video recordings from 2024: LPC 2024 - VFIO/IOMMU/PCI MC.

Older recordings are available through the official YouTube channel of the Linux Plumbers Conference and the archived LPC 2017 VFIO/IOMMU/PCI MC web page at Linux Plumbers Conference 2017, where the audio recordings from the MC track and links to presentation materials are available.

The tentative schedule will provide an update on the current state of VFIO/IOMMU/PCI kernel subsystems, followed by a discussion of current issues related to the proposed topics.

The following was a result of last year's successful Linux Plumbers MC:

```
- The first version of work on solving the complex and pressing issue of secure device assignme
```

Tentative topics that are under consideration for this year include (but are not limited to):

```
- PCI
    - Cache Coherent Interconnect for Accelerators (CCIX)/Compute Express Link (CXL) expansion
    - Data Object Exchange (DOE)
    - Integrity and Data Encryption (IDE)
    - Component Measurement and Authentication (CMA)
    - Security Protocol and Data Model (SPDM)
    - I/O Address Space ID Allocator (IOASID)
    - INTX/MSI IRQ domain consolidation
    - Gen-Z interconnect fabric
    - PCI error handling and management, e.g., Advanced Error Reporting (AER), Downstream Port
    - Power management and devices supporting Active-state Power Management (ASPM)
    - Peer-to-Peer DMA (P2PDMA)
    - Resources claiming/assignment consolidation
    - DMA ownership models
    - Thunderbolt, DMA, RDMA and USB4 security

- VFIO
    - I/O Page Fault (IOPF) for passthrough devices
    - Shared Virtual Addressing (SVA) interface
    - Single-root I/O Virtualization(SRIOV)/Process Address Space ID (PASID) integration
    - PASID in SRIOV virtual functions
    - TDISP/TSM Device assignment/sub-assignment

- IOMMU
    - /dev/iommufd development
    - IOMMU virtualisation
    - IOMMU drivers SVA interface
    - DMA-API layer interactions and the move towards generic dma-ops for IOMMU drivers
    - Possible IOMMU core changes (e.g., better integration with the device-driver core, etc.)
```

If you are interested in participating in this MC and have topics to propose, please use the Call for Proposals (CfP) process. More topics might be added based on CfP for this MC.

Otherwise, join us in discussing how to help Linux keep up with the new features added to the PCI interconnect specification. We hope to see you there!

Key Attendees:

```
 - Alex Williamson
- Benjamin Herrenschmidt
- Bjorn Helgaas
- Dan Williams
- Ilpo Järvinen
- Jacob Pan
- James Gowans
- Jason Gunthorpe
- Jonathan Cameron
- Jörg Rödel
- Kevin Tian
- Lorenzo Pieralisi
- Lu Baolu
- Manivannan Sadhasivam
```

Contacts:

```
 - Alex Williamson (alex.williamson@redhat.com)
- Bjorn Helgaas (helgaas@kernel.org)
- Jörg Roedel (joro@8bytes.org)
- Lorenzo Pieralisi (lpieralisi@kernel.org)
- Krzysztof Wilczyński (kwilczynski@kernel.org)
```

Primary author: WILCZYŃSKI, Krzysztof

Co-authors: WILLIAMSON, Alex; HELGAAS, Bjorn (Google); ROEDEL, Joerg (AMD); PIERALISI, Lorenzo

Presenters: WILLIAMSON, Alex; HELGAAS, Bjorn (Google); ROEDEL, Joerg (AMD); WILCZYŃSKI, Krzysztof; PIERALISI, Lorenzo

Contribution ID: **31** Type: **not specified**

x86 MC

x86-focused material has historically been spread out at Plumbers. This will be an x86-focused microconference. Broadly speaking, anything that might affect arch/x86 is on topic, except where there may be a more focused discussion occurring, like around Confidential Computing or KVM.

This microconference would look at how to address new x86 processor features and also look back at how older issues might be made less painful. For new processor features like APX, what is coming? Are the vendors coordinating, are they coordinating enough so that Linux engineers can talk and are they compatible? For older issues like hardware security vulnerabilities, is the current approach working? If not, how should they be dealt with differently? Can new hardware features or vendor policies help?

As always, the microconference will be a great place for coordination among distributions, toolchains and users up and down the software stack. All the way from guest userspace to VMMs.

Potential Problem Areas to Address:

- Old processor support HIGHMEM64 is gone. What else can be excised from 32-bit? Errata handling, especially when there are microcode fixes
- VFM"infrastructure and new Intel Families
- Expanded x86 GPRs, (aka. APX). How can they get used in the kernel? Any ways to use them while maintaining backward compatibility
- CPU Mitigation work is still painful and error-prone.
- TLB management growing pains: INVLPGB, RAR and the complex dance of home-grown synchronization primitives in switch_mm_irqs_off()
- Microcode issues: Runtime CPU vulnerability updates if mitigated in microcode How is the "old_microcode"mechanism working?
- cpu_feature_enabled(): Friend or Foe?

Key Attendees

- Peter Zijlstra peterz@infradead.org
- Borislav Petkov - bp@alien8.de
- Dave Hansen dave.hansen@linux.intel.com
- Thomas Gleixner tglx@linutronix.de
- Kirill A. Shutemov kirill.shutemov@linux.intel.com LAM and TDX/CoCo topics
- Andrew Cooper andrew.cooper3@citrix.com
- Tom Lendacky thomas.lendacky@amd.com
- H. Peter Anvin h.peter.anvin@intel.com

Primary authors: PETKOV, Boris; HANSEN, Dave

Presenters: PETKOV, Boris; HANSEN, Dave