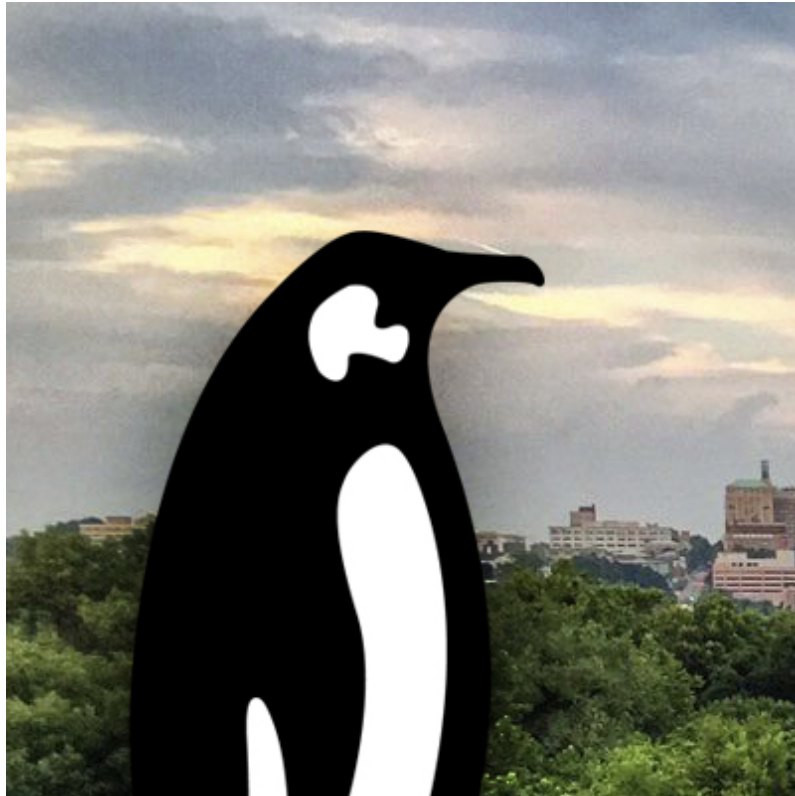# Linux Plumbers Conference 2023

Monday, 13 November 2023 - Wednesday, 15 November 2023

Program

LPC Refereed Track

Refereed presentations are 45 minutes in length (which includes time for questions and discussion) and should focus on a specific aspect of the "plumbing" in the Linux system. Examples of Linux plumbing include core kernel subsystems, core libraries, windowing systems, management tools, device support, container run-times, media creation/playback, and so on. The best presentations are not about finished work, but rather problems, proposals, or proof-of-concept solutions that require face-to-face discussions and debate.

Kernel Summit Track

The goal of the Kernel Summit track will be to provide a forum to discuss specific technical issues. The program committee will also consider "information sharing" topics if they are clearly of interest to the wider development community (i.e., advanced training in topics that would be useful to kernel developers).

We will be reserving roughly half the Kernel Summit slots for last-minute discussions that will be scheduled during the week of Plumber's, in an "unconference style".

Birds of a Feather (BoF)

BoF sessions are free-form get-togethers for people wishing to discuss a particular topic.

eBPF & Networking Track

The track will be composed of talks, 30 minutes in length (including Q&A discussion).

Proposals can cover a wide range of advanced topics related to Linux networking and BPF covering improvements in areas such as (but not limited to) kernel core networking, protocols, routing, performance, tunneling, drivers, BPF infrastructure and its use in tracing, security, networking, scheduling and beyond, as well as non-kernel components like libraries, compilers, testing infra and tools.

Toolchains Track

The purpose of the Toolchains Track is to work on particular problems or issues involving development tools (compilers, assemblers, debuggers, ...) and the Linux kernel. This covers both the GNU toolchain and clang/LLVM.

The aim of the track is to fix particular issues which are of the interest of the kernel, by getting the toolchain developers in touch with the pertinent kernel hackers and by reaching specific and concrete solutions, in situ. It is not about presenting research or abstract/miscellaneous toolchain work.

LPC Microconference

A microconference is supposed to be research and development in action and an abstract for a microconference should be thought of as a set of research questions and problem statements.

In past years microconferences were organized around topics such as security, scalability, energy efficiency, toolchains, containers, printing, system boot, Android, scheduling, filesystems, tracing, or real-time. The LPC microconference track is open to a wide variety of topics as long as it is focussed, concerned with interesting problems, and is related to open source and the wider Linux ecosystem. We are happy about a wide range of topics!

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.

Build Systems MC

In the Linux ecosystems there are many ways to build all the software used to put together a running system. Whether it's building all the binary packages for a binary Linux distribution, using a source-based distribution, or building an embedded system from scratch, there are a lot of shared challenges which each system solves in their own way.

Compute Express Link MC

Compute Express Link is a cache coherent fabric that in recent years has been gaining momentum in the industry. CXL 3.0 launched just before Plumbers 2022 (where very early discussions were had), bringing new challenges such as dynamic capacity devices and large scale fabrics, two features that bring significant challenges to Linux. There has also been controversy and confusion in the Linux kernel community about the state and future of CXL, regarding its usage and integration into, for example, the core memory management subsystem. Many concerns have been put to rest through proper clarification and setting of expectations.

Confidential Computing MC

The Confidential Computing microconferences in the past years brought together developers working secure execution features in hypervisors, firmware, Linux Kernel, over low-level user space up to container runtimes. A broad range of topics was discussed ranging from enablement for hardware features up to generic attestation workflows.

Containers and checkpoint/restore MC

The Containers and Checkpoint/Restore Microconference 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.

Internet of Things MC

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

Kernel Testing & Dependability MC

The Kernel Testing & Dependability Microconference focuses on advancing the current state of testing of the Linux Kernel and its related infrastructure. The main purpose is to improve software quality and dependability for applications that require predictability and trust. We aim to create connections between folks working on similar projects, and help individual projects make progress.

KVM MC

KVM (Kernel-based Virtual Machine) enables the use of hardware features to improve the efficiency, performance, and security of virtual machines created and managed by userspace. KVM was originally developed to host and accelerate "full" virtual machines running a traditional kernel and operating system, but has long since expanded to cover a wide array of use cases, e.g. hosting real time workloads, sandboxing untrusted workloads, deprivileging third party code, reducing the trusted computed base of security sensitive workloads, etc. As KVM's use cases have grown, so too have the requirements placed on KVM and the interactions between it and other kernel subsystems.

Linux Kernel Debugging MC

When things go wrong, we need to debug the kernel. There are about as many ways to do that as you can imagine: printk, kdb/kgdb over serial, tracing, attaching debuggers to /proc/kcore, and post-mortem debugging using core dumps, just to name a few. Frequently, tools and approaches used by userspace debuggers aren't enough for the requirements of the kernel, so special tools are created to handle them: crash, drgn, makedumpfile, libkdumpfile, and many, many others.

Live Patching MC

The Live Patching microconference aims to gather stakeholders and interested parties to discuss proposed features and outstanding issues in live patching.

Power Management and Thermal Control MC

The Power Management and Thermal Control Microconference focuses on power management and thermal control infrastructure, CPU and device power-management mechanisms, and thermal control methods. In particular, we are interested in improving the thermal control infrastructure in the kernel to cover more use cases and utilizing energy-saving opportunities offered by modern hardware in new ways.

Real-time and Scheduling MC

Over the past decade, many parts of PREEMPT_RT have been included in the official Linux codebase. Examples include real-time mutexes, high-resolution timers, lockdep, ftrace, RCU_PREEMPT, threaded interrupt handlers, and more. The number of patches that need integration has been significantly reduced, and the rest is mature enough to make their way into mainline Linux.

The scheduler is 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, where CPU isolation is critical.

RISC-V MC

2023 edition of the RISC-V microconference at Linux Plumbers. Broadly speaking anything related to both Linux and RISC-V is on topic

Rust MC

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.

Tracing MC

The Linux kernel has grown in complexity over the years. Complete understanding of how it works via code inspection has become virtually impossible. Today, tracing is used to follow the kernel as it performs its complex tasks. Tracing is used today for much more than simply debugging. Its framework has become the way for other parts of the Linux kernel to enhance and even make possible new features. Live kernel patching is based on the infrastructure of function tracing, as well as BPF. It is now even possible to model the behavior and correctness of the system via runtime verification which attaches to trace points. There is still much more that is happening in this space, and this microconference will be the forum to explore current and new ideas.

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.

LPC Microconference Proposals

CLOSED