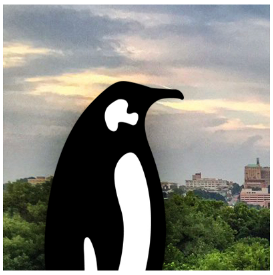

Contribution ID: **8**

Type: **not specified**

Kernel Testing & Dependability MC

The Linux Plumbers 2023 Kernel Testing & Dependability track 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.

This track is intended to promote collaboration between all the communities and people interested in the Kernel testing & dependability. This will help move the conversation forward from where we left off at the LPC 2022 Kernel Testing & Dependability MC.

We ask that any topic discussions focus on issues/problems they are facing and possible alternatives to resolving them. The Microconference is open to all topics related to testing on Linux, not necessarily in the kernel space.

Potential testing and dependability topics:

- KernelCI: Topics on improvements and enhancements for test coverage
- Growing KCIDB, integrating more sources (https://kernelci.org/docs/kcidb/)
- Better sanitizers: KFENCE, improving KCSAN. (https://lwn.net/Articles/835367/)
- 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?
- 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 version?
- Explore how do SBOMs figure into dependability?

Things accomplished from last year:

- Developing a new, modern API for KernelCI with Pub/Sub interface
- Adding Rust coverage in KernelCI https://linux.kernelci.org/job/rust-for-linux/branch/rust/
- KCIDB is continuing to gather results from many test systems: KernelCI, Red Hat's CKI, syzbot, ARM, Gentoo, Linaro's TuxSuite etc. The current focus is on generating common email reports based on this data and dealing with known issues.
- KFENCE is continuing to aid in detecting Out-of-bound OOB accesses, use-after-free errors (UAF), Double free and Invalid free and so on.
- Clang: CFI, weeding out issues upstream, etc.

- Kselftest continues to add coverage for new and existing features and subsystems.
- KUnit is continuing to act as the standard for some drivers and a de facto unit testing framework in the kernel . (https://www.youtube.com/watch?v=78gioY7VYxc)
- The Runtime Verification (RV) interface from Daniel Bristot de Oliveira was successfully merged.

MC Leads:

- Sasha Levin
- Guillaume Tucker
- Shuah Khan

Unconfirmed to-be attendees:

- Sasha Levin
- Kevin Hilman
- Guillaume Tucker
- Alice Ferrazzi
- Veronika Kabatova
- Nikolai Kondrashov
- Antonio Terceiro
- Mark Brown
- Don Zickus
- Enric Balletbo
- Tim Orling
- Gustavo Padovan
- Bjorn Andersson
- Milosz Wasilewski
- Shuah Khan
- Martin Peres
- Arnd Bergmann
- Remi Duraffort
- Peter Zijlstra
- Daniel Stone
- Jan Lübbe
- Dmitry Vyukov
- Brendan Higgins
- Greg KH
- Anders Roxell
- Guenter Roeck
- Jesse Barnes
- Kees Cook
- Kate Stewart
- Sudip Mukherjee
- Daniel Bristot de Oliveira
- Gabriele Paoloni
- Paul Albertella
- Elana Copperman
- Lukas Bulwahn

Primary authors: TOCKER, Guillaume; LEVIN, Sasha; KHAN, Shuah (The Linux Foundation); KHAN, Shuah

Track Classification: LPC Microconference Proposals