

# Linux Plumbers Conference 2025



Contribution ID: 1

Type: **not specified**

## Kernel Testing & Dependability MC

### CFP ends on September 30th (CLOSED)

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