

How to measure kernel testing success.

Wednesday 26 August 2020 10:30 (30 minutes)

Over the years, more services are contributing to the testing of kernel patches and git trees. These services include Intel's 0-day, Google's Syzkaller, KernelCI and Red Hat's CKI. Combined with all the manual testing done by users, the linux kernel should be rock solid! But it isn't.

Every service and tester is committed to stabilizing the linux kernel, but there is duplication and redundant testing that makes the testing effort inefficient.

How do we know new tests are filling in the kernel gaps? How do we know each service isn't running the same test on the same hardware? How do we measure this work towards the goal of stabilizing the linux kernel?

Is functional testing good enough?

Is fuzzing good enough?

Is code coverage good enough?

How to incorporate workload testing?

How to leverage the unified kernel testing data (kcidb)?

This talk is an open discussion about those problems and how to address them. I encourage maintainers to bring ideas on how to qualify their subsystem as stable.

By the end of the talk, a core set of measurables should be defined and trackable on kernelci.org with clear gaps that testers can help fill in.

I agree to abide by the anti-harassment policy

I agree

Primary author: ZICKUS, Don (Red Hat)

Presenter: ZICKUS, Don (Red Hat)

Session Classification: Testing and Fuzzing MC

Track Classification: Testing and Fuzzing MC