

# Data-race detection in the Linux kernel

*Monday 24 August 2020 10:00 (45 minutes)*

In this talk, we will discuss data-race detection in the Linux kernel. The talk starts by briefly providing background on data races, how they relate to the Linux-kernel Memory Consistency Model (LKMM), and why concurrency bugs can be so subtle and hard to diagnose (with a few examples). Following that, we will discuss past attempts at data-race detectors for the Linux kernel and why they never reached production quality to make it into the mainline Linux kernel. We argue that a key piece to the puzzle is the design of the data-race detector: it needs to be as non-intrusive as possible, simple, scalable, seamlessly evolve with the kernel, and favor false negatives over false positives. Following that, we will discuss the Kernel Concurrency Sanitizer (KCSAN) and its design and some implementation details. Our story also shows that a good baseline design only gets us so far, and most important was early community feedback and iterating. We also discuss how KCSAN goes even further, and can help detect concurrency bugs that are not data races.

Tentative Outline:

- Background
- What are data races?
- Concurrency bugs are subtle: some examples
- Data-race detection in the Linux kernel
- Past attempts and why they never made it upstream
- What is a reasonable design for the kernel?
- The Kernel Concurrency Sanitizer (KCSAN)
- Design
- Implementation
- Early community feedback and iterate!
- Beyond data races
- Concurrency bugs that are not data races
- How KCSAN can help find more bugs
- Conclusion

Keywords: testing, developer tools, concurrency, bug detection, data races

References: <https://lwn.net/Articles/816850/>, <https://lwn.net/Articles/816854/>

## I agree to abide by the anti-harassment policy

I agree

**Primary author:** ELVER, Marco (Google)

**Presenter:** ELVER, Marco (Google)

**Session Classification:** LPC Refereed Track

**Track Classification:** LPC Refereed Track (Closed)