Contribution ID: **39** Type: **not specified**

Rust MC

Rust is a systems programming language that is making great strides in becoming the next big one in the domain.

Rust for Linux aims to bring it into the kernel since it has a key property that makes it very interesting to consider 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.

This microconference intends to cover talks and discussions on both Rust for Linux as well as other non-kernel Rust topics.

Possible Rust for Linux topics:

- Bringing Rust into the kernel (e.g. status update, next steps…).
- Use cases for Rust around the kernel (e.g. drivers, subsystems…).
- Integration with kernel systems and infrastructure (e.g. wrapping existing subsystems safely, build system, documentation, testing, maintenance…).

Possible Rust topics:

- Language and standard library (e.g. upcoming features, memory model…).
- Compilers and codegen (e.g. rustc improvements, LLVM and Rust, rustc_codegen_gcc, gccrs…).
- Other tooling and new ideas (Cargo, Miri, Clippy, Compiler Explorer, Coccinelle for Rust…).
- Educational material.
- Any other Rust topic within the Linux ecosystem.

I agree to abide by the anti-harassment policy

Primary authors: OJEDA, Miguel; ALMEIDA FILHO, Wedson

Presenters: OJEDA, Miguel; ALMEIDA FILHO, Wedson

Track Classification: LPC Microconference Track (CLOSED)