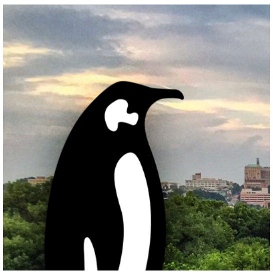

Contribution ID: **24**

Type: **not specified**

Rust

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

Rust for Linux is the project adding support for the Rust language to the Linux kernel. Rust has a key property that makes it very interesting 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. It also provides other important benefits, such as improved error handling, stricter typing, sum types, pattern matching, privacy, closures, generics, 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:

- Rust in the kernel (e.g. status update, next steps…).
- Use cases for Rust around the kernel (e.g. subsystems, drivers, other modules…).
- Discussions on how to abstract existing subsystems safely, on API design, on coding guidelines…
- Integration with kernel systems and other infrastructure (e.g. build system, documentation, testing and CIs, maintenance, unstable features, architecture support, stable/LTS releases, Rust versioning, third-party crates…).
- Updates on its subprojects (e.g. klint, pinned-init…).

Possible Rust topics:

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

Last year was the first edition of the Rust MC and the focus was on showing the ongoing efforts by different parties (compilers, Rust for Linux, CI, eBPF…). Shortly after the Rust MC, Rust got merged into the Linux kernel. Abstractions are getting upstreamed, with the first major drivers looking to be merged soon: Android Binder, the Asahi GPU driver and the NVMe driver (presented in that MC).

Primary authors: OJEDA, Miguel; ALMEIDA FILHO, Wedson

Track Classification: LPC Microconference Proposals