



Rust for CORE-V

Philipp Krones



What is Rust?

- Rust is a high performant and safe systems programming language
- People love Rust
 - Performance
 - Memory safety without GC
 - Powerful type system (static strong typing with inference)
 - Modern conveniences
 - Community
- Multiparadigm
 - Imperative, structured, functional, concurrent, generic, compiled
- Free and open-source software, MIT or Apache 2.0 License

Who is using Rust?

- **Mozilla** - Servo, Stylo, WebRender.
- **Google** - Fuchsia operating system, crosvm.
- **Facebook** - Libra, Mononoke, Mercurial and HHVM.
- **Microsoft** - Azure IoT, VSCode, other things.
- **Amazon** - Firecracker.
- **Dropbox** - Storage system.

You can see even more familiar names like **ARM, Twitter, npm, Red Hat, Reddit, Samsung, Cloudflare, Gnome, Chef, Canonical, Coursera, Tor** and many more.

Being used also for **Blockchain, Networking, Embedded, Games, WebAssembly, Native Extensions, Web APIs**, etc.

Memory safety

	C/C++	Rust
Control, flexibility	✓	✓
Minimal to no runtime	✓	✓
Double free	✗	✓
Use after free	✗	✓
Data race	✗	✓

Modern Tooling

- **rustup**: Use multiple Rust versions for different directories
- **cargo**: Automatically download, build, and link dependencies
 - `cargo build`, builds your whole project. No `Makefile`, no `cmake`, ...
- **rustfmt**: Format Rust code according to style guidelines
- **clippy**: Additional warnings for dangerous or unidiomatic code
- **rust-analyzer**: Implementation of the Language Server Protocol
- **Documentation**: Great documentation for the whole ecosystem

Support Tiers

- Tier 1
 - Official binary releases are provided for the platform.
 - Automated **testing** is set up to run tests for the platform.
 - Documentation for how to use and how to build the platform is available.
- Tier 2
 - Official binary releases are provided for the platform.
 - Automated **building** is set up, but may not be running tests.
- Tier 3
 - Supported in the Rust codebase, but not built automatically
 - ⇒ CORE-V support would start here

<https://doc.rust-lang.org/nightly/rustc/platform-support.html>

Support Tiers

- Tier 1
 - `x86_64-pc-windows-msvc`
 - `x86_64-unknown-linux-gnu`
 - `x86_64-apple-darwin`
- Tier 2
 - `riscv32{i,imac,imc}-unknown-none-elf`
 - `riscv64{gc,imac}-unknown-none-elf`
 - `riscv64gc-unknown-linux-gnu`
- Tier 3
 - `riscv32gc-unknown-linux-gnu`
 - `CORE-V?`

<https://doc.rust-lang.org/nightly/rustc/platform-support.html>

Rust support for CORE-V

1. LLVM has to support CORE-V
2. Tell rustc that LLVM supports CORE-V
3. (Get CORE-V to Tier 2 support level)