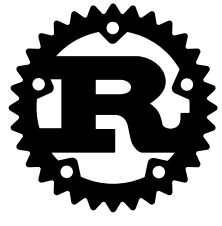
Rust for Safety



Intro

Tyler Neely

- https://github.com/spacejam
- makes and breaks databases and distributed systems
- <3 high performance + high correctness

ferrous systems

Florian Gilcher

- https://github.com/skade
- Rubyist
- Rust community and core teams
- Rust Trainer since 2015

ferrous systems

Ferrous Systems

- Consultancy making Rust better for everyone
- Open Source library development
- Finding modern FOSS funding models

OpenCollective

- https://opencollective.com/ferrous-systems-gmbh
- rust-analyzer: a new Rust IDE
- sled: fast and concurrent database
- async-rs: an asynchronous port of the rust stdlib with a modern scheduler

Trainings

- 3 days base trainings
- 2 day conferences
- Rust for embedded
- Rust for async programming

Masterclasses:

- High performance threading
- Rust ergonomics

You!

- What do you want to use Rust for?
- Which are the questions that you've always been interested in?
- If you already worked with Rust: what was the hardest part for you?
- With which of the following languages do you identify most?
  - C/C++
  - Objective-C/Swift
  - Haskell/ML
  - Ruby/Python/Perl/Java

Course goals

In this workshop, you will:

- Get a solid understanding of the basics of Rust
- Learn Safe Rust
- Understand Rust concurrency guarantees
- Understand the role of unsafe Rust
- Understand Rust lifetimes
- Get a glimpse into the possibility Rust opens

Out of scope

- FFI
- (Advanced) Generics
- `.async/.await`

Course Structure

- Theory talks
- Exercises
- Spotlight talks to highlight interesting approaches
- Open Q&A

Exercises

- Building a small protocol parser
- Testing the parser with `proptest`
- Building a small server accepting the protocol
- Make the server threaded/concurrent

Curiousity

Ask questions! Many of them! As early as possible!