Rust

Why it might be worth a look

Presenters

Uwe Arzt: mail@uwe-arzt.de

Kai Kratz: k.kratz@fz-juelich.de

TOC

- Motivation
- When to use Rust
- Getting started
- Some language highlights
- Integrations
- Learning
- Ask your questions

Motivation

We would like you to have a look at Rust and consider it for your next project instead of (espacially but no limited) C / C++.

Rust has great potential for all fields, where correctness and speed are key goals.

Many people would like to program in Rust, see: https://insights.stackoverflow.com/survey/2021#most-loved-dreaded-and-wanted-language-love-dread

When to use Rust

- Runtime Performance
- Low memory overhead
- Parallelism
- Zero Cost Abstractions
- If you need fast **and** correct memory handling
  - C++: fast, but hard to manage (look into Nicos Talk about move semantics) https://www.youtube.com/watch?v=PNRju6_yn3o
  - Java: Garbage collected, easy but sometime hard to predict

Whenever C++ was the choice, but i need memory safety and correctness.

Getting started

- rustup -> A rust toolchain / tool management tool
    - install platform and cross compilers (c++ apt / python apt)
- cargo -> build tool
    - create project
    - build project
    - test project
- rustc -> your compiler
    - c++ g++, clang, msvc, intel / python 2.7 / 3.x

Batteries included

- Dependency Management
    - Cargo has a complete dependency managment with a central registry
    - All crates on crates.io is also used for testing new rust releases
- rustdoc
    - C++ doxygen + breath + Sphinx / python: Sphinx
    - Code inside Doc is tested too
- rustfmt -> auto format your code
    - C++ clang-format, IDE / python: black,...
- clippy -> additional lints
    - C++ clang-tidy, cppcheck / python: pep8, pylint
- rustrls -> brings autocompletion and code navigation to your IDE

Highlights 1

enum

Variables can be embedded into enums

```
enum Color {
    None,
    Gray(u8),
    Color(u8, u8, u8),
}
```

tuples

```
let (ret1, ret2) = function();

fn function -> (u8, u8) {
  (12, 34)
}
```

Highlights 2

match (switch on steroids)

```rust
match (x, y) {
  (1, 1) => println!("both one"),
  (1, a) if a % 2 == 0 => println!("first parameter one, second even"),
  (1, _) => println!("first parameter one"),
  (_, _) => println!("everything else"),
}
```

Result type

```rust
fn ex_result(fail: bool) -> Result<u8, String> {
  if fail {
    return Err("xxx".to_string());
  }
  Ok(7)
}
```

Language Integration

- You can use C and C++ libraries from rust. There is also a integration to build C / C++ libraries with Cargo
- You can use rust libraries from Python, C and C++ and everything that can use a C library (using the C calling conventions)
- The integration of Rust for Linux Kernel Driver Development has started

Some Python libs are already written in Rust, i.e. cryptography.

IDE support

- vscode (we have already seen some examples)
- All kinds of Editors have Rust support (vim, sublime)
- Many IDE integration (CLion, IntelliJ, PyCharm, Eclipse, ...)

Learning Material I

Books

- The Rust Book https://doc.rust-lang.org/book/
  In print: https://nostarch.com/Rust2018
- Rust for Rustaceans https://rust-for-rustaceans.com

Documentation

- Rust Standard Library Documentation https://doc.rust-lang.org/std/
- Official overview of Rust learning material https://www.rust-lang.org/learn
- Rust cheat sheets https://cheats.rs/

Learning Material II

Talks / Videos

- A firehose of Rust
  https://www.youtube.com/watch?v=IPmRDS0OSxM
- A firehose of Rust (extended)
  https://www.youtube.com/watch?v=FSyfZVuD32Y
- Some more advanced talks and live coding sessions
  https://www.youtube.com/c/JonGjengset
- Introducing Rust features
  https://www.youtube.com/c/LetsGetRusty

Learning Material III

Community

- Short tour of Rust
  https://tourofrust.com/
- Weekly Rust Newsletter
  https://this-week-in-rust.org/
- Rust playground (run snippets)
  https://play.rust-lang.org/
- Rust discord

Learning Material IV

Other

- Many commandline tools are rewritten in Rust
  https://zaiste.net/posts/shell-commands-rust/
- Use Rust from Python:
  https://github.com/PyO3/pyo3
  https://dygalo.dev/blog/rust-for-a-pythonista-3/
- Compiler Explorer for Rust (see Assembly from the compiler)
  https://rust.godbolt.org/

Infos/Questions

presentation can be found on https://codeberg.org/uwearzt/rust-presentation.git

Ferris