Concepts of programming languages

Rust

Tobias van Driessel, Tom Freijsen, Ruben Schenkhuizen, Floris Schild, Mats Veldhuizen

Universiteit Utrecht

Presentation Schedule

▶ Background
▶ Design principles
▶ What problems does Rust solve and how?
▶ Practical details

Universiteit Utrecht

Background

- Personal project of Mozilla employee
- Sponsored by Mozilla Research
- First stable release (1.0) in 2015
- Used in Firefox and by Dropbox
- Most loved programming language - SO Developer Survey

Universiteit Utrecht

Quick Overview

▶ Statically typed
▶ Functional **and** Imperative paradigms
▶ Strict language

Universiteit Utrecht

Beautiful Quotes

Mozilla Research:

"Rust is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety."

"It fuses the expressive and intuitive syntax of high-level languages with the control and performance of a low-level language."

Universiteit Utrecht

[Faculty of **Science**
Information and Computing Sciences]

Universiteit Utrecht

Features

▶ zero-cost abstractions

Features

▶ zero-cost abstractions

▶ move semantics

Universiteit Utrecht

Features

▶ zero-cost abstractions

▶ move semantics

▶ guaranteed memory safety

Universiteit Utrecht

Features

▶ zero-cost abstractions

▶ move semantics

▶ guaranteed memory safety

▶ threads without data races

Universiteit Utrecht

Features

▶ zero-cost abstractions

▶ move semantics

▶ guaranteed memory safety

▶ threads without data races

▶ trait-based generics

Universiteit Utrecht

Features

▶ zero-cost abstractions

▶ move semantics

▶ guaranteed memory safety

▶ threads without data races

▶ trait-based generics

▶ pattern matching

Universiteit Utrecht

Features

- ▶ zero-cost abstractions

- ▶ move semantics

- ▶ guaranteed memory safety

- ▶ threads without data races

- ▶ trait-based generics

- ▶ pattern matching

- ▶ type inference

Universiteit Utrecht

Features

- ▶ zero-cost abstractions
- ▶ move semantics
- ▶ guaranteed memory safety
- ▶ threads without data races
- ▶ trait-based generics
- ▶ pattern matching
- ▶ type inference
- ▶ minimal runtime

Universiteit Utrecht

[Faculty of **Science**
Information and Computing Sciences]

Features

▶ zero-cost abstractions

▶ move semantics

▶ guaranteed memory safety

▶ threads without data races

▶ trait-based generics

▶ pattern matching

▶ type inference

▶ minimal runtime

▶ efficient C bindings

Universiteit Utrecht

AHHH ALLEMAAL STUFFFF

Universiteit Utrecht

Practical

▶ Mature & intuitive package manager (cargo)

▶ Lots of libraries from a vibrant community

▶ i.a. IDE priority this year

▶ Basic linting and debugging

▶ Get started on https://doc.rust-lang.org

▶ Used because of safety, performance and being practical

Universiteit Utrecht

[Faculty of **Science**
Information and Computing Sciences]