

Rust : Introduction



David Bouchet

david.bouchet.epita@gmail.com

Why Rust?

Low-Level

Less safe
More Control
Faster

Assembly C++

C

Usually Compiled

High-Level

Safer
Less control
Slower

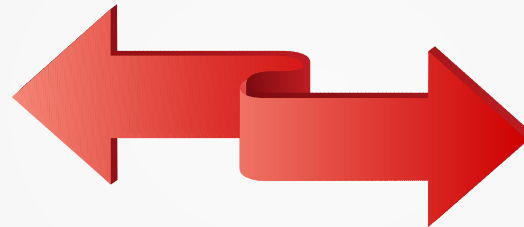
Go Java

C#

Python Ruby

PHP JavaScript

Usually Interpreted



Why Rust?

Why are C and C++ unsafe?

- Buffer Overflows
- Uncontrolled Format Strings
- Memory Leaks
- Null Dereferencing
- Dangling Pointers
- Race Conditions
- Array Index out of Bounds

Why Rust?

Why are high-level languages slower?

Because they use an *interpreter* and/or a *garbage collector*.

These mechanisms make the execution of a program **safer** and **prevent memory leaks**.

They also provide **strong abstraction from the hardware**. It is then **easier and faster to develop a program**.

But unfortunately, this comes with a runtime cost.

Why Rust?

What is Rust?

“**Rust** is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety.” (1)

“Its design lets you create programs that have the performance and control of a low-level language, but with the powerful abstractions of a high-level language.” (2)

“**Rust** is a programming language that’s focused on safety, speed, and concurrency.” (2)

(1) <https://www.rust-lang.org/en-US/index.html>

(2) <https://doc.rust-lang.org/book/second-edition/ch01-00-introduction.html>

Why Rust?

Featuring ⁽¹⁾

- Zero-Cost Abstraction
- Move Semantics
- Guaranteed Memory Safety
- Threads Without Data Races
- Trait-Based Generics
- Pattern Matching
- Type Inference
- Minimal Runtime
- Efficient C Bindings

(1) <https://www.rust-lang.org/en-US/index.html>

Resources

- **Official Website:**

<https://www.rust-lang.org/>

- **Rust Playground:**

<https://play.rust-lang.org/>

- **The Rust community's crate registry:**

<https://crates.io/>

- **MDN web docs (Mozilla)**

<https://developer.mozilla.org/en-US/docs/Mozilla/Rust>