

Rust: Benefits And Weaknesses

H.Assaran



What is Rust?

- ▶ A systems programming language designed for memory safety and performance.
- ▶ Combines the speed and control of C/C++ with the safety guarantees of languages like Java.
- ▶ Developed by Mozilla Research and used in major projects like Firefox and the Linux kernel.

Who is Rust founder?

- ▶ Graydon Hoare
- ▶ Started in 2006 in Mozilla
- ▶ First version released in 2010
- ▶ Achieved a winning streak, emerging as the "most loved programming language" in the Stack Overflow Developer Survey for three consecutive years: 2016, 2017, and 2018.



Why is Rust called Rust?

- ▶ It was named after a group of remarkably hardy fungi that are “over-engineered for survival.”



Benefits of Rust

- ▶ **Memory safety:** Rust's ownership system and compiler prevent memory errors, leading to more stable and secure software.
 - ▶ Microsoft estimates that 70% of the vulnerabilities in its code are due to memory errors from code written in C/C++ languages.
- ▶ **Performance:** Achieves performance comparable to C/C++ without sacrificing safety.
- ▶ **Concurrency:** Rust's design helps prevent data races and other concurrency issues in multithreaded applications.
- ▶ **Modern Features:** Rust offers various modern features like generics, closures, and pattern matching, making it expressive and concise.
- ▶ **Zero-cost abstractions:** Avoids the performance penalty of traditional abstractions.
- ▶ **Low Energy Consuming:** It needs half as much electricity as a similar program written in Java
- ▶ **Friendly Community:** The Rust community is known for being welcoming and helpful, with resources and support available for newcomers.
- ▶ **Integrated Package Manager:** Cargo, Rust's built-in package manager, simplifies dependency management.

Weaknesses of Rust

- ▶ **Steep learning curve:** Borrowing and ownership concepts require significant understanding.
- ▶ **Compile-time errors:** The strictness can lead to many compile-time errors, requiring careful attention to detail.
- ▶ **Slower Compile Time:** Rust's compiler can be slower than those of some other languages, which can impact development speed.
- ▶ **Limited library support:** Compared to older languages, the ecosystem is still growing.
- ▶ **Not as beginner-friendly:** Requires some programming experience to grasp its nuances.

Is Rust right for you?

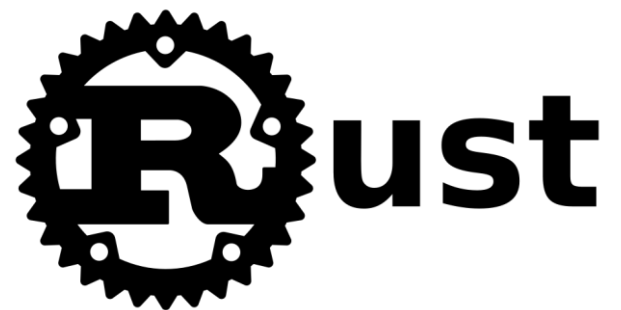
- ▶ **Consider your project's requirements:**
 - ▶ Does performance and memory safety matter at the same time?
 - ▶ Do you need robust concurrency features?
 - ▶ Are you comfortable with a steeper learning curve?
- ▶ **Evaluate your team's skills and experience:**
 - ▶ Do you have developers familiar with Rust concepts?
 - ▶ Are you willing to invest in learning the language

Rust: not New Idea but Practical

- ▶ Tricks employed weren't new ideas. But rust engineers were adept at finding these well-honed concepts and turning them into practical, usable features.

Rust: A language for the future?

- ▶ Growing popularity
- ▶ Unique combination
- ▶ Addressing key challenges



Conclusion

- ▶ Rust is a powerful and promising language with a unique blend of features.
- ▶ Its safety, performance, and expressiveness make it attractive for various applications.
- ▶ However, the learning curve and limited ecosystem might not suit every project.
- ▶ While Rust's core ideas weren't new, the Rust team's skill lay in making these established concepts practical and user-friendly.
- ▶ Rust is a language for the future due to its growing popularity, unique combination of features, and effective addressing of key challenges in modern software development.
- ▶ Carefully evaluate your needs and resources before adopting Rust.

Areas for next presentations

- ▶ Comparing Rust with C, C++, Java, Python, Go
- ▶ Borrowing and Ownership in Rust