描述

Rust 是什么?

A language empowering everyone to buildreliable and efficient software

一种使每个人都能建立可靠和高效的软件的语言(系统级编程语言)

特性

高性能:速度极快且内存效率极高，没有运行时或垃圾收集器，它可以为性能关键的服务提供动力，在嵌入式设备上运行，并容易与其他语言集成

可靠性:丰富的类型系统和所有权模型保证了内存安全和线程安全(能够在编译时消除许多类错误)

生产力: Rust 有很棒的文档、一个带有友好错误信息提示的编译器和一流的工具 (一个集成的包管理器(cargo)和构建工具、具有自动完成和类型检查的智能多编辑器支持、一个自动格式化程序等等)

Wiki: 在计算机科学中，垃圾回收 (GarbageCollection，简称 GC) 是指一种自动的存储器管理[81机制。当某个程序占用的一部分内存空间不再被这个程序访问时，这个程序会借助垃圾回收算法向操作系统归还这部分内存空间。垃圾回收器可以减轻程序员的负担，也减少程序中的错误。垃圾回收最早起源于 LISP语言。Java、C#、Go和D 语言都支持垃圾回收器

生态

Command Line- 利用 Rust 强大的生态系统快速开发出CLI工具。Rust 可以帮助你可靠地维护应用程序，并轻松地发布它WebAssembly- 使用 Rust 来增强你的JavaScript，每个 rust crate 都对应一个js 模块可以发布到npm，用前端打包工具 (webpack.vite、rollup等)进行捆绑

Networking- 可预测的性能。小5的资源占用。稳如磐石的可靠性。Rust是网络服务的最佳选择(构建高性能服务器)

Embedded- 针对低资源的设备?需要低层次的控制而不放弃高层次的便利? Rust可以满足你的要求(嵌入式开发，内核开发等)

Rust助力前端--Rust 因其出色的表现，连续7年在 StackOverflow调查中成为最受欢迎的语言 (Stack OverflowDeveloper Survey 2022) 。优秀的特性自然也受到了前端的青睐，很多大神开始基于 Rust 重写前端生态

个人理解，不准确的地方欢迎讨论: Rust 之所以优秀是因为它借鉴了大量编程语言的优秀特性，通过其自身的类型系统，生命周期，所有权等概念将其统一起来，成为其安全可靠，高性能的重要保证(构建速度暂时不吐槽，毕竟团队也在不断努力提速中)。在编译阶段会对代码进行检查，抛出友好的错误，并给出大量指导性修改建议 (很多时候都是编译器在教我写代码)，这在别的语言似乎很难看到。