

Use Rust To Make A TSDB

- Rust Feature
- Rust vs C/C++
- Why to choose Rust to make a TSDB
- Rust 'hello world'
- Rust Course Schedule
- Rust Learning Experience Sharing

Rust Feature

- Rust定位于一门系统级编程语言，没有历史的包袱，集各家语言之长
- Rust 生态社区 快速发展中。
- Rust 提供内存安全、线程安全而又不失性能的现代化系统级编程语言。
- Rust有完全不亚于C++的表达能力和性能，又解决了C++的最大痛点

Rust Feature

- 2006年，Rust作为Graydon Hoare的个人项目首次出现。
- 2009年，Graydon Hoare成为[Mozilla](#)雇员[\[14\]](#)。
- 2010年，Rust首次作为Mozilla官方项目出现[\[15\]](#)。
- 2011年，Rust成功的完成了移植[\[17\]](#)。Rust的自编译器采用[LLVM](#)作为其编译后端。
- 2012年1月20日，第一个有版本号的[预览版](#)Rust编译器发布[\[18\]](#)。
- 2015年5月16日，Rust 1.0.0发布[\[32\]](#)。
- 2021年2月8日，[AWS](#)、[华为](#)、[Google](#)、[微软](#)以及Mozilla宣布成立Rust基金会

Rust Feature

- 变量的可变性
- 所有权与借用
- 生命周期
- 无空指针
- Trait与范型
- 异步
- 包管理系统
- 测试模块

Rust vs C/C++

- 预习的结果?
- 想重点关注的地方?

Rust vs C/C++

- Option与空指针
- 没有实现Copy trait的对象，绑定、赋值、非引用传参时默认是移动语义
- 变量默认是不可变绑定，需要修改变量则需用可变绑定
- 支持函数表达式返回
- 支持零尺寸的结构体、空枚举、空数组 (`[T, 0]`) 两种字符串类型变量：
- 使用match模式匹配 代替 `switch & case` 与 `c++`的`std::visit`类似
- 支持`?`运算符，函数返回异常时，返回对应的错误`Err<T>`
- 链式调用
-

Why Rust

- 同学来回答一下?

Rust ‘hello world’

- rust 安装

mac/linux:

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

win:

x86_64-pc-windows-msvc (官方推荐)

- 编辑器

vscode + Rust and Friends (插件集合)

clion + rust 插件

Course Schedule

- rust 基础 二 三 四 五
变量/类型系统/struct/生命周期 模式匹配 多态/traits
智能指针 错误处理 常见集合 测试用例
- rust 进阶 六 七 八 九
内存布局 异步与并发 性能分析 leetcode
- rust 结课
在用rust 写下一代的时序数据库 Cnosdb
介绍一下 我们的项目代码和设计思路
欢迎大家积极参与 一键三连

Rust Learning Experience Sharing

- 昨夜西风凋碧树，独上高楼，望尽天涯路。
- 衣带渐宽终不悔，为伊消得人憔悴。
- 众里寻他千百度，蓦然回首，那人却在，灯火阑珊处。

Rust Learning Experience Sharing

《Rust 程序设计语言》

<https://rust.bootcss.com/title-page.html>

《Rust 高级编程 》

<https://doc.rust-lang.org/nomicon/intro.html>

《Rust语言圣经》

<https://book.rust.team/>

Rust Learning Experience Sharing

- <https://github.com/RustMagazine>
- <https://github.com/RustStudy>
- <https://github.com/rust-unofficial>
- <https://www.resource.rs/>
- <https://rustwiki.org/>
- <https://users.rust-lang.org/>
- <https://rustcc.cn/>
- <https://wiki.mozilla.org/Rust>

Q & A