Rust 编程笔记

adamanteye

1. 文档及参考

1.1. 手册

- Rust 单页手册

1.2. 代码规范

- Rust API Guidelines

1.2.1. 命名

`crate` 的命名应当为 `snake_case`,更多讨论参考 Naming convention for crates · rust-lang/api-guidelines · Discussion 29

1.3. 性能

- The Rust Performance Book

2. 调试技巧

dbg!宏用于打印到 `stderr`. 可以在 `cargo test` 下使用,也可以调试 release 构建下出现的问题.

3. 异步

参考:

- Async Rust 的实现 | 夜天之书

4. 超好用的库

4.1. 错误处理

- dtolnay/anyhow: Flexible concrete Error type built on std::error::Error

4.2. 输入输出

- kkawakam/rustyline: Readline Implementation in Rust
- Nukesor/comfy-table: Build beautiful terminal tables with automatic content wrapping
- clap-rs/clap: A full featured, fast Command Line Argument Parser for Rust

4.3. 异步

无脑 tokio 全家桶就对了:

- tokio-rs/tokio: A runtime for writing reliable asynchronous applications with Rust. Provides I/O, networking, scheduling, timers, ...

支持并发读写的容器:

- xacrimon/dashmap: Blazing fast concurrent HashMap for Rust

4.4. 并行

多线程并行化:

- rayon-rs/rayon: Rayon: A data parallelism library for Rust

4.5. 内存分配

- purpleprotocol/mimalloc_rust: A Rust wrapper over Microsoft's MiMalloc memory allocator

4.6. 解析器

- Nom — Parser
- zesterer/chumsky: Write expressive, high-performance parsers with ease

4.6.1. chumsky

`chumsky` 是一个 PEG 解析器.

- `choice` 选择第一个成功匹配的,注意安排优先级
- `to_slice` 会把整个解析器的输入范围都捕获为原始字符串,所以应该放在可能有的 `then_ignore` 之前

4.7. 网页服务器

自己一直习惯用 `axum` 来着,虽然上手会费劲一点,但感觉逻辑还是很自洽的,其他著名的库还有 `actix-web`, `poem`, `rocket`,前者开发不顺,后两者没有试过.

此外,支持 OpenAPI 规范的 `utoipa` 也是简化文档编写的利器.

关于后端服务性能调优,可以参考:

- Structural changes for +48-89% throughput in a Rust web service – Sander Saares