



Rust 学习经验分享

河南科技大学-徐堃元

2024/4/7



主要内容

- 个人学习经验
- Rust 语言特性
- 为何使用 Rust 进行系统编程
- Rust 在操作系统开发中的应用
- 学习资源与社区支持
- 结论与展望
- 致谢



个人学习经验

- 初步接触：2023年春夏季开源操作系统训练营。
- 我的学习路径：从基础语法到系统编程实践。
- 遇到的挑战：理解所有权、借用和生命周期的概念。
- 解决方法：多写代码，多查资料。
- 建议：针对语法的学习尽量扎实，不然在后续的系统编程实践中会遇到较大的阻力。



Rust 语言特性

- 所有权系统：独特的所有权模型确保内存安全。
- 借用和生命周期：编译时检查引用的有效性。
- 错误处理：使用Result和Option类型进行显式错误传播。
- 模块和包管理：良好的模块化支持和包管理工具。
- 宏系统：Rust的宏系统允许元编程，使得可以写出更抽象和复用的代码。



为何使用 Rust 进行系统编程

- 内存安全：所有权系统防止了空悬指针和数据竞争。
- 性能优异：零成本抽象，无垃圾收集器。
- 并发性：支持无数据竞争的并发编程模型。
- 现代化语言特性：模式匹配、类型推导、泛型等。
- 完善的工具链与裸机编程支持。



Rust 在操作系统开发中的应用

- rCore: 使用 Rust 编写的类Unix操作系统。
- ArceOS: 使用 Rust 编写的组件化操作系统。
- vivo buleOS。
- ArceOS Hypervisor 虚拟化技术。
- 基于 Rust 异步编程技术的异步操作系统开发。
- Rust for Linux 驱动。



学习资源与社区支持

- 开源操作系统训练营。
- 官方网站：提供最新资讯和文档。
- 《Rust 语言圣经》
- 《Rust 程序设计语言》：官方发布的权威指南。
- Rust by Example：实用的代码示例集合。
- Rust中文社区等在线社区：问题解答和讨论。



结论与展望

- Rust作为一种现代系统编程语言，为操作系统开发带来了新的可能性。
- 通过学习和实践，可以深入理解Rust的特性和优势。
- 展望未来，Rust可能会在更多领域得到应用和发展。



致谢

- 感谢大家的聆听和参与!
- 如果有任何问题, 欢迎在群内讨论或者向我提问。