

LITTLEFS的RUST实现

清华大学计算机系



答辩学生：段津荣



指导老师：向勇

目录

CONTENTS



01

研究目标回顾

02

已有研究成果

03

后续研究计划

01

研究目标回顾

1.1 核心目标

LittleFS 是一个为嵌入式系统设计的轻量级文件系统，专注于低存储开销、低资源消耗和高效的闪存存储管理。满足嵌入式设备对于文件系统的要求，包括支持随机断电、考虑块磨损等特性

用Rust语言重写该文件系统可以做到：

- 1.保持性能高效
- 2.提高安全性，避免了空指针解引用、缓冲区溢出和数据竞争等常见内存错误
- 3.有更好的抽象设计，易于模块化编程和扩展
- 4.为嵌入式设备提供rust原生实现的文件系统

1.2 开题计划表

寒假-第2周

学习rust语言和rcore，熟悉rust开发操作系统流程
学习littleFS的设计文档

第5-6周

初步完成目标文件系统的rust实现，并整理相关文档，进行中期检查

9-11周

撰写论文文稿，测试文件系统，整理和挂载实验记录和项目文档

第3-4周

完成目标文件系统的设计和元数据、CTZ跳表等核心数据结构的实现

第7-8周

调试和改进目标文件系统，使之具有可用性，并封装为ArceOS模块和通用crate

12-15周

逐周打磨定稿，以及时间冗余

02

已有研究成果

1.1 文档成果

- 1.翻译并学习littlefs_DESIGN和littlefs_SPEC两篇技术文档，翻译结果合计共两万字
- 2.学习并分析littlefs源码，撰写三篇代码分析技术文档，针对源码中全部的宏、枚举、结构体、函数作出注解，合计共十万字（包含源码本身）。
- 3.撰写多篇篇研发日志和周报，记录研究进展，合计约七千字
- 4.以上学习、研究过程产生的文字记录，均收录、整合至网站<https://duanjr.github.io/GraduationProjectRecords/>

1.2 代码成果

- 1.完成目标文件的全部数据结构的设计和实现（约1000行）
- 2.设计并实现AI workflow，结合语言大模型完成littlefs中全部函数的rust实现（约8000行）。但尚未排除全部错误，尚不能良好运行。
- 3.产出十余个工作脚本（约2000行）、多个针对littlefs的不同版本（原版实现，低级别绑定的rust实现，毕设目标实现）的测试代码（约2000行）
- 4.上述代码均整合整理仓库<https://github.com/duanjr/GraduationProject>

03

后续研究计划

3.1 进度自检

- 1.完全、足额的实现开题报告的研究计划目标
- 2.留下数量充足、高质量、完全公开的文字记录

3.2 计划调整

随着研究的实际展开，开题时定下的计划与目标存在不合理指出，需要调整：

- 1.研究重点应该是完整、高质量的实现易于调用的littlefs的rust实现。这种情况下，封装为ArceOS等实际使用场景的模块不应是毕设内容的一部分，在目标文件系统合理完成的情况下，封装方应该仅需非常小的工作就能够使用此文件系统
- 2.内部设计本希望参考[littleFS2](#)，但经研究， littleFS2只是一个密码产品公司内部使用的文件系统，且littleFS的设计结构和功能和std::fs差之甚远，把它设定为目标高级接口不是一个好的选择， littleFS2事实上也并没有完全实现std::fs的结构和接口。我认为更合理的目标实现是[fuser](#)所设计的[FileSystem 特征](#)，是更合理的rust实现设计。
- 3.增加在ESP8266开发板、星光板等物理设备上运行和测试通过的实现目标

谢谢 THANKS



清华大学
Tsinghua University



答辩学生：段津荣



指导老师：向勇

