

//// 基于RUST语言和嵌入式系 统的故障安全文件系统

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

研究背景

02

研究目标

03

研究计划



01

研究背景

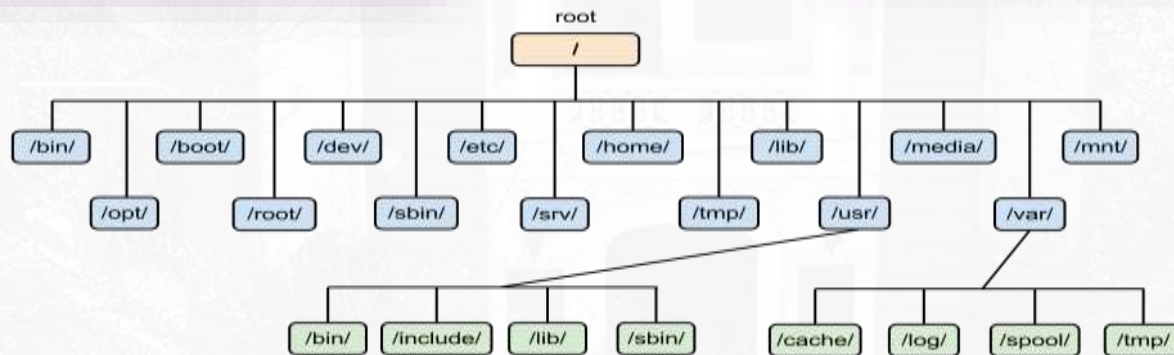
1.1 文件系统

文件系统

计算机中的文件系统是一种存储和组织计算机数据的方法，它使得对数据的访问和查找变得容易，文件系统使用文件和树形目录的抽象逻辑概念代替了硬盘和光盘等物理设备使用数据块的概念，用户只需要记住这个文件的所属目录和文件名。

不同的使用场景对所使用的文件系统有不同的特性要求，例如：

- 高性能计算要求文件系统提供极快的IO性能和并行处理能力
- 分布式文件系统要求文件系统进行数据分片，并支持高并发和高容错
- 大文件的处理需要文件系统支持更高的大小上限和高效地管理组织数据



1.2 嵌入式设备对文件系统的要求

资源有限

嵌入式系统通常具备资源有限的硬件（如低功耗、有限存储等），这要求文件系统需要严格控制对内存的使用并尽可能地轻量

随机断电

由于嵌入式设备应用场景的复杂性，更容易面临简单和不稳定的电源环境，这要求文件系统提供断电恢复功能。

块磨损

嵌入式设备常用闪存进行写入，更具有破坏性，这要求文件系统考虑到块磨损并设计动态的均衡磨损策略以防止设备过早损坏。

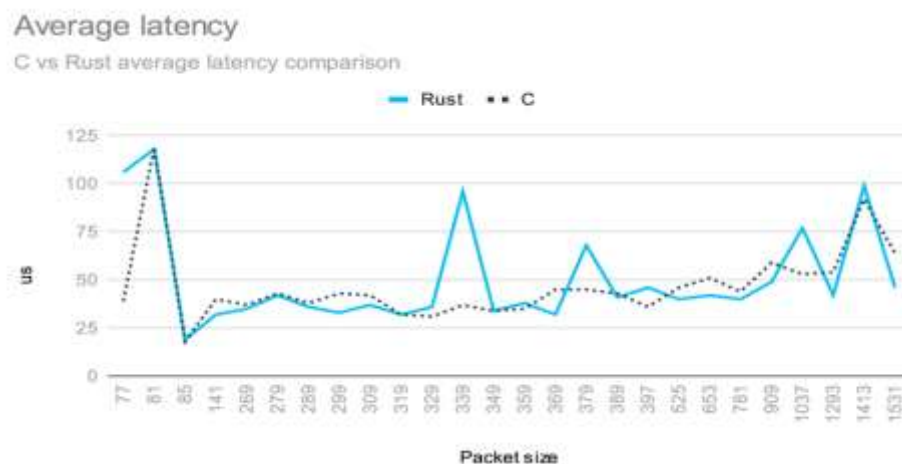
1.3 基于rust语言的操作系统内核

优势

- 非常高效，与C的性能不相上下
- 安全性高：内存管理安全，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
- 有更好的抽象能力，易于模块化编程

有关成果与需求

- ArceOS：应用场景广泛的模块化操作系统
- Embassy：嵌入式应用框架，支持编写更加高效、安全的异步嵌入式应用
- 均缺乏适配于嵌入式设备的文件系统



02

研究目标

2.1 概述

参考C语言实现的littleFS文件系统，使用rust语言实现适用于嵌入式设备的故障安全文件系统，提供断电恢复、受限RAM下工作、动态磨损等特性。实现完成后依次封装为通用crate和可接入模块化单内核操作系统ArceOS的模块。

2.2 设计参考

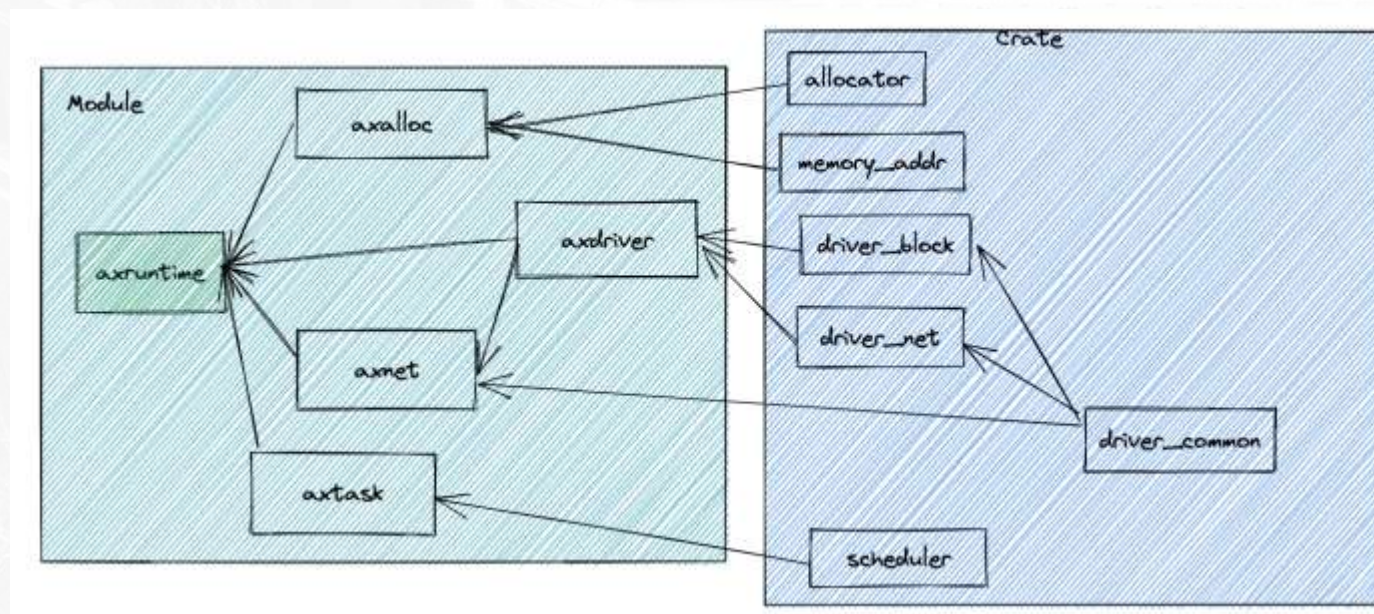
LittleFS 是一个为嵌入式系统设计的轻量级文件系统，专注于低存储开销、低资源消耗和高效的闪存存储管理。参考LittleFs的设计思路，我们可以确定目标文件系统的相关特性：

- 结合日志文件系统和COW文件系统的优势，达到低内存开销，低传播磨损，原子更新的特性目标
- 在文件元数据中通过使用两个块的循环日志结构实现断电恢复、避免数据损坏风险
- 使用CTZ跳表作为COW数据结构，使得文件数据的读写操作在常量内存下高效进行。
- 使用动态磨损均衡功能的块分配器

2.3 封装使用

ArceOS: 在模块化ArceOS操作系统中, 目标文件系统将被封装为crate层的一个模块, 并提供上层VFS所规定的各项接口

LittleFS2: LittleFS2是封装littleFS, 提供rust API的。目标文件系统也将封装为与LittleFS2具有相同api的crate, 并配备相关文档。[扩展: 为Embassy框架做异步]



03

研究计划

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

初步实现具有完整功能的目标文件系统, 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, 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

谢谢 THANKS



清华大学
Tsinghua University



答辩学生：段津荣



指导老师：向勇

