

春夏季开源操作系统学习与工作总结

汇报人：周屿涵

2025 年 6 月 21 日

目录

- 1 学习历程回顾
- 2 近期工作总结 (5.31 - 6.21)
- 3 未来工作计划
- 4 总结与感悟

学习：第一阶段：核心理论与 Rcore 实践

构建坚实的操作系统知识体系

- **Rust 语言深化：**

- 系统性掌握所有权、借用及生命周期等内存安全核心机制，为内核开发奠定语言基础。

- **操作系统内核探索：**

- **内核架构：**梳理了从系统引导、模块加载到协同工作的完整流程。
- **进程管理：**深入理解进程状态变迁、上下文切换与 CPU 调度策略。
- **内存管理：**掌握了物理内存分配、虚拟地址映射及多级页表等关键技术。

学习：第二阶段：ArceOS 动手实践与挑战

在实践中检验与深化理论知识

- **基础功能实现：**

- 为系统增添了彩色打印、哈希映射等基础支持模块，提升了系统的易用性与功能性。

- **核心挑战攻克：**

- **Shell 功能增强：**基于现有系统调用，巧妙地实现了 `mv` 命令。
- **内存动态映射：**成功实现了 `sys_map` 系统调用，支持内存区域的灵活映射。
- **缺页异常处理：**在 `rcore` 基础上，为 ArceOS 扩展并实现了 `page_fault` 处理机制。
- **虚拟化技术初探：**通过对关键寄存器的配置，成功搭建了 `simple_hv` (基础 Hypervisor) 原型。

工作：内核库优化：int_ratio

提升代码质量与测试覆盖率

核心机制

int_ratio 是一个面向 no_std 环境的高性能整数比例计算库，其核心思想是通过预计算，将高开销的除法操作转换为高效的乘法和位移操作。

代码质量提升

- 解决了所有 Clippy 警告。
- 遵循标准规范化文档注释。
- 统一代码及文档标记，提升可读性与可维护性。

测试覆盖增强

- 建立了独立的集成测试套件。
- 全面覆盖等效性、逆运算、精度等核心场景。
- 重点验证了零值、边界条件和 Panic 等异常行为的正确性。

工作：文件系统组件：axfs 修复与测试

保障模块化设计的健壮性与可靠性

核心理解：VFS、RAMFS 与 DEVFS 的协同

- `axfs_vfs`: 虚拟文件系统层，定义了一套标准的文件系统接口 (Trait)，作为统一的抽象。
- `axfs_ramfs`: 基于内存的文件系统，是对 VFS 接口的一种通用、可读写的具体实现。
- `axfs_devfs`: 特殊的设备文件系统，用于将 `/dev/null` 等设备抽象为文件。

主要贡献：构建全面的集成测试

新增的集成测试系统性地验证了 `vfs` 的路径解析、`devfs` 的设备 I/O 以及 `ramfs` 从创建、读写到删除的完整生命周期，显著提升了文件系统的稳定性。

现有组件增强与演进

axfs_ramfs (内存文件系统) 打磨

- **性能优化**: 探索分块存储以高效支持大文件和稀疏文件; 将全局锁细化为粒度更小的节点锁, 提升并发性能。
- **功能增强**: 引入资源配额 (Quota) 管理机制; 研究并实现文件系统状态的持久化与恢复。

axfs_devfs (设备文件系统) 丰富

- **扩充设备生态**: 实现 `/dev/random`、`/dev/urandom` 等标准虚拟设备。
- **实现动态管理**: 支持在系统运行时动态地注册和注销设备。

认领新任务与拓展学习

计划认领新组件：kernel_guard

- **核心功能理解：**

- 一个遵循 RAIL 设计模式的内核同步原语。
- 用于自动管理临界区，进入时自动禁用中断或抢占，退出时自动恢复。

- **个人目标：**

- 将已学知识应用到内核同步这一全新领域，拓展技术视野。
- 借此机会深入理解操作系统内核的并发控制与同步机制。

总结与感悟

- **个人背景：**电子科技大学成都学院，大二在读。
- **成长与蜕变：**
 - 作为二战训练营的“老兵”，我经历了从去年面对 Rcore 无从下手，到如今能深入 ArceOS 源码、主动贡献高质量测例并规划未来优化的转变。
- **方法论：脚踏实地，仰望星空**
 - 坚持从最基础的组件优化和测试工作做起，在实践中夯实基础。
 - 以此为基石，持续深入研究文件系统，目标是逐步参与到更核心的内核开发中。
- **展望：学无止境，持续前行**
 - 这段经历是宝贵的起点，我将以更大的热情投入后续的学习与开发，持续为开源社区贡献力量。

感谢观看