

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 学习历程总结
- ② 工作总结 (5.31 - 6.21)
- ③ 未来计划
- ④ 总结与感悟

第一阶段：操作系统核心理论与 Rcore 实践

- **深入 Rust 语言：**
 - 全面掌握其所有权、借用和生命周期等内存安全核心机制。
- **深探操作系统内核：**
 - **内核架构：**理解从系统启动到模块交互的全过程。
 - **进程管理：**掌握进程生命周期与 CPU 调度算法。
 - **内存管理：**研究物理内存分配、虚拟内存映射及页表机制。

第二阶段：ArceOS 实验与挑战

将理论知识应用于动手实践

- **实现基础系统功能：**

- 完成了彩色打印、哈希映射支持等基础模块。

- **攻克核心挑战任务：**

- **Shell 增强：**基于现有接口，巧妙实现 `mv` 命令。
- **内存映射：**完成 `sys_map` 系统调用，实现内存区域的灵活映射。
- **缺页处理：**在 `rcore` 基础上，扩展并实现了 `page_fault` 处理机制。
- **虚拟化初探：**通过修改关键寄存器，成功搭建了 `simple_hv` (基础 Hypervisor)。

内核组件优化：int_ratio 库

提升代码质量与测试覆盖率

核心理解

int_ratio 是一个为 no_std 环境设计的高性能整数比例计算库，通过预计算将除法转为高效的乘法与位移。

代码质量提升

- 解决所有 Clippy 警告。
- 规范化文档注释。
- 统一标记格式，提升文档可读性。

测试覆盖增强

- 创建独立的集成测试。
- 覆盖等效性、逆运算、精度差异等场景。
- 确保了零值、边界值和 Panic 行为的正确性。

文件系统组件修复与测试：axfs

深入理解并保障其模块化设计的健壮性

核心理解：VFS、RAMFS、DEVFS 的关系

- `axfs_vfs`: 定义文件系统必须遵守的“接口标准” (Trait)。
- `axfs_ramfs`: 通用、可读写的“内存文件系统”实现。
- `axfs_devfs`: 用于表示特殊设备（如 `/dev/null`）的“设备文件系统”实现。

主要贡献：添加了全面的集成测试

系统性地验证了 `vfs` 的路径处理、`devfs` 的设备 I/O 行为以及 `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和可靠性。

组件增强与功能演进

axfs_ramfs (内存文件系统) 的打磨

- **性能优化**: 引入分块存储以支持大文件和稀疏文件; 将全局锁细化为节点锁, 提升并发性能。
- **功能增强**: 增加资源配额 (Quota) 管理; 实现状态的持久化与恢复。

axfs_devfs (设备文件系统) 的丰富

- **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**: 支持在运行时动态添加和移除设备。

认领新任务与持续学习

计划认领新组件：kernel_guard

- **初步理解：**

- 它是一个遵循 RAIL 模式的工具，用于在内核中创建临界区。
- 能自动处理中断禁用/使能，或内核抢占的禁止/允许。

- **目标：**

- 将所学知识应用到内核同步这一新领域。
- 深入理解操作系统内核的并发控制与同步机制。

- **个人背景：**电子科技大学成都学院大二的学生。
- **一名“老兵”的坚持：**
 - 作为一名二战训练营的成员，我经历了从去年对 Rcore 感到无从下手，到如今能深入 ArceOS 源码、添加测例并规划改进的成长。
- **脚踏实地，仰望星空：**
 - 最基础的工作做起，持续研究文件系统，目标是能一步步参与到更核心的开发工作中。
- **学无止境，持续奋斗：**

感谢观看