加分项：FS 操作 – SSD 友好的 FS

中国科学院大学 操作系统研讨课

2017.12.20

1. 任务要求

- 实现任务一中可选实现的接口，包括 utime chmod destory readlink
- 基于任务一，设计 SSD friendly 的文件系统，参考 F2FS 实现日志式文件系统(LFS)，包括日志式布局、后台垃圾回收（GC）。

1.1. 需要了解的部分

SSD 特性：

- 寿命有限
- 擦后写，以及商品固态盘因此引入的地址转换、垃圾回收、预留空间，本实现亦可根据直接操作 NAND 的方式设计，即没有地址转换层 FTL。
- 读延迟 X0us <<写延迟 X ms ≈ 擦除延迟 X ms
- 并行性，内部有多个通道

2. 初始代码

2.1. 文件介绍

- common.h/.c 定义文件系统的基本数据结构，以及文件操作接口，请将代码添加到对应文件。
- disk.h/.c 提供访问 sd 接口，详情见头文件。
- logging.h/.c 提供 log 功能接口，详情见头文件。
- fuse-main.c 程序 main，请补全 fuse_ops。
- lib/fuse.h fuse 头文件，不包含在 p6 代码中，仅供查询参考，请自行安装并使用 fuse。

2.2. 获取

课程网站。

2.3. 运行

Makefile 文件提供编译功能。

3. 任务

3.1. 设计和评审

帮助学生发现设计的错误，及时完成任务。学生需要对这次的作业进行全面 考虑，在实现代码之前有清晰的思路。学生讲解设计思路时可以用不同的形式， 如伪代码、流程图

等，建议使用 PPT。

3.1.1. 设计介绍

请详细说明你的针对固态盘的文件系统设计，相比于经典机械磁盘文件系统有何优点？

3.2. 开发

3.2.1. 要求

在 common.h 中实现文件系统所需的数据结构，包括 superblock inode 、dentry 、file handle 以及你所设计的内存目录索引结构。

在 common.c 中实现文件系统接口，按要求实现必须实现部分，根据情况自行选择是否实现可选部分。

在 fuse-main 中补全相应部分，并实现 mount 功能。

实现可选接口

- Utime 修改文件的 access time， modify time
- Chmod 修改文件权限
- Destroy 释放文件系统内存结构，用于 umount 时调用
- Readlink 读取软链接文件的数据
-

实现 LFS 式文件系统：包括异地更新的数据布局和异步的垃圾回收。可以参考作业提供的 paper“F2FS: A New File System for Flash Storage”

- 异地更新指的是，考虑到 SSD 擦后写的特性，在文件系统分配数据块时以追加分配的方式进行分配，当文件数据发生修改时，给文件分配新的数据块，而不是直接修改数据块，同时将旧数据块标记为无效。Log-structured 的文件系统将数据块组织成循环日志，将文件的随机写转换成顺序写，有助于利用 SSD 内部的并行性，提高性能。
- 文件系统定期在这个循环日志中，回收由于异地更新被标记为无效的数据的过程称作垃圾回收。如果 LFS 是以在日志尾追加分配的方式分配的，则 LFS 的空闲数据块为日志尾到日志头的数据块，当数据块小于阈值（自定）时，需要触发垃圾回收。垃圾回收自日志头开始扫描，拷贝有效数据到新的位置，同时回收无效数据，最后移动日志头。

3.2.2. 注意事项

4. 任务测试

针对所需实现的接口，验证功能正确性。

设计并验证 LFS 的布局实现及垃圾回收工作的正确性

参考资料

[1] Changman Lee, Dongho Sim, Joo-Young Hwang, and Sangyeun Cho, F2FS: A New File

System for Flash Storage, FAST 2015■