

任务一：基于 FUSE 的文件系统

中国科学院大学 操作系统研讨课

2017.12.20

1. 任务要求

在 Linux 操作系统 (x86 架构) 中, 基于 FUSE 实现用户态的文件系统 p6fs, 要求该文件系统能够在 Linux 中管理 SD 卡, 并能正确挂载到一个挂载点进行文件、目录等操作。

设计并实现文件系统元数据, 数据块索引与布局(layout), 并至少支持间接(2 级)索引:

- 文件系统元数据结构: superblock, inode, dentry, file_handle。
- 数据块分配。
- 数据块索引。
- 目录的组织形式。
- 文件系统磁盘布局, 至少包括 2 份 superblock。

设计并实现内存目录项索引结构, 例如链表、哈希、B+树等。

设计并实现 FUSE 文件系统接口, 标为可选实现的接口请自行选择是否实现。

设计并实现文件系统的 mkfs 和 mount 功能 (p6fs_ops->init)。

1.1. 需要了解的部分

用户态文件系统 FUSE(Filesystem in Userspace)工作原理。FUSE 的目的是让用户在不修改内核的情况下, 在用户态实现自己的文件系统。FUSE 包括内核模块与 libfuse 两部分, 用户的文件操作经 VFS 到内核 FUSE 模块, 然后交给用户态文件系统程序执行, 执行完成后原路返回。

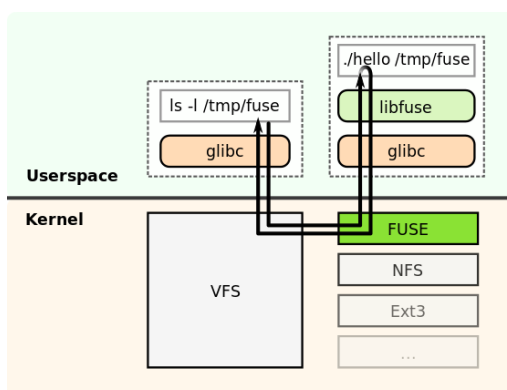


图 1 FUSE 工作流程图

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。
- example/hello.c fuse 提供的 example code。

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。

请**必须**准备文件系统数据布局与索引的设计图示, 及目录的组织方式与内存索引结构。根据你的设计, 提供单个文件最大尺寸计算说明。

3.1.1. 设计介绍

- Superblock 数据结构, 需要包含什么内容? 有什么用? 多份 superblock 如何保证一致?
- inode 数据结构, 需要保存什么信息? 是否有读写并发考虑?
- 数据块管理, 数据块索引、分配、空闲块信息维护方式。
- 目录的组织形式、目录项索引结构设计情况, 路径中的当前路径"."与父目录".."如何处理?
- 文件系统各个接口的设计, 如 mknod, link, open 等
- 文件系统 init 与 mount 如何工作?

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。

表 1 p6 必须实现的接口

fs 管理	init	初始化 fs,包括 mount 与 mkfs
	statfs	打印文件系统统计信息, 对应 shell df 命令
目录操作	mkdir	创建目录, 同 shell 命令
	rmdir	删除目录, 同 shell 命令
	readdir	打印目录下的内容, 对应 shell ls 命令
文件操作	mknod	创建文件, 可使用 shell touch 命令
	symlink	符号链接, 可使用 shell ln -s 命令
	link	硬链接, 可使用 shell ln 命令
	unlink	删除链接/删除文件, 可使用 shell rm 命令
	open/release	打开/关闭文件, 支持 read-only, write-only, read-write
	read/write/truncate	读/写/截断 文件
	getattr	文件或目录信息打印, 可使用 shell 命令
	rename	包含重命名与移动, 可使用 shell mv 命令

3.2.2. 注意事项

3.2.2.1. 要求注意事项

- p6fs 不要求实现扩展属性 xattr, 与权限相关操作。
- superblock 至少应为 2 份。
- 建议实现 destory 接口以防内存泄露

3.2.2.2. 完成注意事项

- init 需要处理 mount 与 mkfs
- mknod 与 mkdir 中需要解析路径, 同时 mkdir 创建 dentry 时需要处理 "." 与 ".."
- readdir 使用 filler(buf, name, NULL, 0), 其中 name 为 file 名字, NULL 为 <sys/stat.h> 中的 struct stat, 此处可直接填 NULL, 0 为 next entry 地址, 用于目录项保序, 如使用此参数, 需要全程维护 offset, 此处直接填 0。详见 fuse.h。filler 的 buffer 为 fuse 传递, 不需要自己进行内存分配, readdir 遍历目录下目录项时, 将 dentry 中的 filename 作为参数循环调用 filler 即可, 当内存不足时, filler 会返回 1, 当目录较大时, 应注意内存不足的判断。
- read/write 调用 disk.h 接口, disk.h 中定义了磁盘操作接口, device_read/write_sector, 参数为 buffer 与 sector_id, 其中 buffer 需要自己分配, sector 为磁盘块 4K 粒度组织时的 blk_id。
- logging.h/c 中定义了日志接口, 会将操作日志打印到日志文件中, 便于调试。默认打印级别为 DEBUG, 使用使可以直接在 fuse 接口中调用接口, log 的格式与 printf

相同, 即 `DEBUG("xxxx %d",data)`。

- `statfs` 接口中返回项请参考 `<sys/statvfs.h>` `<bits/statvfs.h>`, 包括 `f_bsize` 文件系统数据块大小, `f_frsize` 基本数据块大小, 本任务同 `f_bsize`, `f_blocks` 总共数据块数目, `f_bfree` 空闲块数, `f_bavail` 可以给用户文件分配的数据块数目 (除去文件系统元数据占用), `f_files` inode 数目, `f_ffree` 空闲 inode 数目, `f_favail` 同 `f_ffree`。
- `fuse_file_info`, `fuse` 封装的文件描述符, 在 `fuse` 源码的 `include/fuse_common.h` 中定义。主要包括 `int flags` 成员, 为 `fopen` 的 `flag` 标记 `O_RDONLY` 等, `uint64_t fh` 用于指向 `common.h` 中 `file_info` 结构, 即 `p6fs_open` 分配文件描述符后, 存在 `fuse_file_info->fh` 中。详见参考文献 1
- `getattr` 的 `stat` 相关内容见参考文献 2, 要注意 `stat` 权限相关设置, 用于 `chmod` 接口实现。
- 权限相关 `uid,gid` 可以通过全局变量获取 `struct fuse_context *fuse_con = fuse_get_context();`
- 由于是纯用户态文件系统, 测试磁盘仅作为数据盘, 不可作为启动盘, 磁盘数据块布局时, 可以不为 `bootblock` 预留空间。

4. 任务测试

测试 1 自行准备测试程序验证 `mount` 正确性, 应包括

- case1 正常 `mount`
- case2 破坏其中一份 `superblock` 时 `mount`
- case3 破坏全部 `superblock` 时 `mount`

测试 2 文件系统接口健全测试 (`sanity test`), 包括接口正确性测试, 及边界条件如单文件最大大小、目录下最多目录项数目测试

测试 3 基于 `filebench` 的性能测试, `filebench` 安装请参考《Project 6 预备知识》

参考资料

[1] `statvfs`

http://www.qnx.com/developers/docs/6.5.0/index.jsp?topic=%2Fcom.qnx.doc.neutrino_lib_ref%2Fs%2Fstatvfs.html&cp=13_4_22_162

[2] `stat`

http://www.qnx.com/developers/docs/6.5.0/index.jsp?topic=%2Fcom.qnx.doc.neutrino_lib_ref%2Fs%2Fstatvfs.html

