

超级简单的 Fuse 文件系统 ufs 实验报告

曾千洋 2018011372

运行

ufs 文件系统基于C语言实现，实现源代码全部在 `fuse-proj01/ufs.c` 中。

运行环境为 WSL2, Ubuntu-18.04。

```
PS C:\Users\Sefr> wsl --list --verbose
  NAME          STATE          VERSION
* Ubuntu-18.04  Running        2
```

运行脚本在 `fuse-proj01/script/test.sh` 中。

```
set -e
set -o xtrace

gcc -Wall ufs.c `pkg-config fuse3 --cflags --libs` -o ufs

rm /tmp/disk # mark it if /tmp/disk not exist
touch /tmp/disk
fallocate --length=2g /tmp/disk

rm -rf mount # mark it if mount not exist
mkdir mount
./ufs -d -f mount
```

功能

实现思路

设计文件元数据 `inode`，将文件元数据与文件内容都存入一个模拟磁盘文件 `/tmp/disk` 当中。

为了简单起见，本项目的文件系统是一个扁平化的工作系统，没有实现多层级的目录结构，只有一个根目录，其他的文件和目录均存在根目录当中，节省了工作量，当然也跑不了一些基本的项目了（比如我的期中 KV-Store 项目）。

基本接口

注册并实现了如下的 fuse 文件系统 API：

```

1023 static struct fuse_operations operations = {
1024     .init      = do_init,
1025     .getattr   = do_getattr,
1026     .readdir   = do_readdir,
1027     .mkdir     = do_mkdir,
1028     .mknod    = do_mknod,
1029     .rmdir     = do_rmdir,
1030     .unlink    = do_unlink,
1031     .rename    = do_rename1,
1032     .read      = do_read,
1033     .write     = do_write,
1034 };
1035

```

在 `main` 函数的开始，负责初始化磁盘的不同区域。磁盘文件 `/tmp/disk` 按照块（`BSIZE=4096`）对齐，分为 inode 区域（0-1023 Blocks，存取文件元数据）、bitmap 区域（1024-1031 Blocks，存储file data区域中的块是否有效）、file data区域（1032-262144 Blocks，存储文件内容）。同时，也初始化内存当中的 `inode`。

`getattr` 函数获取文件的属性，最重要的属性就是文件类型是 `directory` 还是 `file`。

`readdir` 函数获取目录下的所有文件名。本项目当中，只允许对根目录应用该函数，因此就是遍历根目录下的所有文件信息，获取相应的文件名。

`mkdir`、`mknod` 函数创建一个新目录/新文件，`rmdir`、`unlink` 函数删除一个空目录/文件。需要更新对应的磁盘区域。

`rename` 函数重命名一个文件/目录。

`read` 函数读取一个 `file` 文件的内容；`write` 函数写入一个 `file` 文件的内容。

实现功能

```
root@LAPTOP-TV7ONE4E:~/fuse-proj01# cd mount
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# ls
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# mkdir dir0
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# ls
dir0
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# cd dir0
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount/dir0# cd ../
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# mkdir dir1
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# rmdir dir0
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# ls
dir1
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# echo Hello >> 1.txt
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# cat 1.txt
Hello
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# cp 1.txt 2.txt
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# ls
1.txt 2.txt dir1
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# mv 2.txt 007.txt
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# cat 007.txt
Hello
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# echo Hi >> 1.txt
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# cat 1.txt
Hello
Hi
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# rm 1.txt
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount# ls
007.txt dir1
root@LAPTOP-TV7ONE4E:~/fuse-proj01/mount#
```

由于没有实现多层级的功能，所以这些功能都只能在项目根目录下执行。

fuse端的输出（只截取了最后的一小段）：

```
    unique: 142, success, outsize: 144
unique: 144, opcode: OPEN (14), nodeid: 4, insize: 48, pid: 17451
    unique: 144, success, outsize: 32
unique: 146, opcode: WRITE (16), nodeid: 4, insize: 83, pid: 17451
write[0] 3 bytes to 6 flags: 0x8401
    write[0] 3 bytes to 6
    unique: 146, success, outsize: 24
unique: 148, opcode: RELEASE (18), nodeid: 4, insize: 64, pid: 0
    unique: 148, success, outsize: 16
unique: 150, opcode: LOOKUP (1), nodeid: 1, insize: 46, pid: 5184
LOOKUP /1.txt
getattr[NULL] /1.txt
    NODEID: 4
    unique: 150, success, outsize: 144
unique: 152, opcode: OPEN (14), nodeid: 4, insize: 48, pid: 5184
    unique: 152, success, outsize: 32
unique: 154, opcode: RELEASE (18), nodeid: 4, insize: 64, pid: 0
    unique: 154, success, outsize: 16
unique: 156, opcode: LOOKUP (1), nodeid: 1, insize: 46, pid: 5227
LOOKUP /1.txt
getattr[NULL] /1.txt
    NODEID: 4
    unique: 156, success, outsize: 144
unique: 158, opcode: UNLINK (10), nodeid: 1, insize: 46, pid: 5227
unlink /1.txt
    unique: 158, success, outsize: 16
unique: 160, opcode: FORGET (2), nodeid: 4, insize: 48, pid: 0
FORGET 4/7
DELETE: 4
unique: 162, opcode: OPENDIR (27), nodeid: 1, insize: 48, pid: 5237
    unique: 162, success, outsize: 32
unique: 164, opcode: GETATTR (3), nodeid: 1, insize: 56, pid: 5237
getattr[NULL] /
    unique: 164, success, outsize: 120
unique: 166, opcode: READDIRPLUS (44), nodeid: 1, insize: 80, pid: 5237
readdirplus[0] from 0
    unique: 166, success, outsize: 656
unique: 168, opcode: LOOKUP (1), nodeid: 1, insize: 45, pid: 5237
LOOKUP /dir1
getattr[NULL] /dir1
    NODEID: 3
    unique: 168, success, outsize: 144
unique: 170, opcode: LOOKUP (1), nodeid: 1, insize: 48, pid: 5237
LOOKUP /007.txt
getattr[NULL] /007.txt
    NODEID: 5
    unique: 170, success, outsize: 144
unique: 172, opcode: READDIR (28), nodeid: 1, insize: 80, pid: 5237
    unique: 172, success, outsize: 16
unique: 174, opcode: RELEASDIR (29), nodeid: 1, insize: 64, pid: 0
    unique: 174, success, outsize: 16
|
```