存储系统课程编程作业——实验报告

一、内容

在 Linux EXT4 文件系统环境下，随机构建五层目录

在第二层目录中进行相关操作，实现 Linux shell 指令中的 ls、ls -l、touch、mkdir、rmdir、mv、chmod

二、前期调查与分析

1、通读内核文档有关 ext4 的部分（https://www.kernel.org/doc/html/v6.1/filesystems/ext4/），大致了解 ext4 metadata 的组织。我需要在 block group 内解析 superblock 和 group descriptor，来拿到 inode 表、了解文件系统中的目录结构。其中，根目录的 inode 号为 2，这个数字是预定义的，从而可以拿到根目录的 inode 结构体。

2、作业要求实现的操作主要是文件夹的增删改查；唯一一个涉及文件的 touch 实际上是空文件，只需要从 inode 表中搞一个新的 inode，不需要动其他 block。本次作业难点在操纵 inode、增改文件数据（因为文件夹就是文件，修改文件数据的能力是必要的 ）、及解析修改文件夹内容上。

3、文件夹也是个文件，文件的内容就是列表本身。为了拿到文件夹中的列表，我要解析这个文件。我要知道文件是怎样存在 block 中的，inode 中的 i_block 很重要，是文件逻辑 block 到磁盘 block 的映射。

4、本次作业工作量最大的关于 metadata 的增删改查，为其设计合适接口——包括拿到 superblock、拿到 inode、拿到 block descriptor、bitmap 操作、分配与释放 block、分配与释放 inode、文件逻辑地址转换成磁盘地址、遍历文件夹的 dentry、删除与增加文件夹的 dentry。上述操纵 metadata 的接口实现之后 ,实现 ls、touch、mkdir、rmdir、mv、chmod 轻而易举。

5、考虑到该作业的体量，我事先 mkfs.ext4 格式化一个 raw 文件；挂载在我目前的电脑上，构建五层目录。而后我用我自己的程序解析这个 raw 文件的 metadata，来实现作业要求的几个命令。为了简化问题，使用位宽 32bit，关闭所有 ext4 高级特性；程序也没有做成交互式 shell，几个命令实现成 C 语言函数。

三、源码

源码中包含三个文件，ext4.h、halo.c、new_ext4.sh。

ext4.h 是从内核中复制来的 ext4 metadata 数据结构。

halo.c 是 C 实现的 ls、ls -l、touch、mkdir、rmdir、mv、chmod 操作，这个程序接受一个 ext4 文件系统的 raw 文件。

new_ext4.sh 把一个 raw 文件格式化成 ext4，然后构建 5 个嵌套的文件夹。

1、首先运行 new_ext4.sh 搞一个 ext4 文件系统的 raw 文件

```
liuweinan@liuweinan-xmu-pc:~/20230505_Storage_system$ ./new_ext4.sh
+ dd if=/dev/zero of=img.ext4 bs=1M count=256
256+0 records in
256+0 records out
268435456 bytes (268 MB, 256 MiB) copied, 0.353476 s, 759 MB/s
+ mkfs.ext4 -q -b 4096 -o Lites -O none img.ext4
+ mkdir -p rootfs
+ sudo mount img.ext4 rootfs
+ cd rootfs
+ sudo mkdir -p 1/2/3/4/5
+ cd ..
+ tree -L 5 rootfs
rootfs
├── 1
│   └── 2
│       └── 3
│           └── 4
│               └── 5
└── lost+found  [error opening dir]

6 directories, 0 files
+ sudo umount rootfs
```

就是先 dd 一个 256mB 的全 0 文件，叫 img.ext4，然后 mkfs.ext4，然后 mount 到一个目录里创建文件夹。

2、编译 halo.c

```
liuweinan@liuweinan-xmu-pc:~/20230505_Storage_system/hw$ gcc halo.c
liuweinan@liuweinan-xmu-pc:~/20230505_Storage_system/hw$
```

直接 gcc 就行，我写这个仅仅依赖 C 库，只涉及到 raw 文件操作。

3、运行试试

```
liuweinan@liuweinan-xmu-pc:~/20230505_Storage_system/hw$ ./a.out img.ext4
Open ext4 img: img.ext4
> ls
.
..
lost+found
1
> cd 1
> ls -l
drwxr-xr-x      3       0       0       4096    1683382865      .
drwxr-xr-x      4       0       0       4096    1683382865      ..
drwxr-xr-x      3       0       0       4096    1683382865      2
> touch 2
2 already exists.
> touch new_file
> ls -l
drwxr-xr-x      3       0       0       8192    1683382865      .
drwxr-xr-x      4       0       0       4096    1683382865      ..
drwxr-xr-x      3       0       0       4096    1683382865      2
-rw-r--r--      1       0       0       0       0       new_file
> mkdir new_dir1
> mkdir new_dir2
> ls -l
drwxr-xr-x      5       0       0       16384   1683382865      .
drwxr-xr-x      4       0       0       4096    1683382865      ..
drwxr-xr-x      3       0       0       4096    1683382865      2
-rw-r--r--      1       0       0       0       0       new_file
drw-r--r--      2       0       0       4096    0       new_dir1
drw-r--r--      2       0       0       4096    0       new_dir2
> rmdir new_dir1
> rmdir new_file
new_file is not dir
> rmdir 2
dir 2 has content
```

```
> ls -l
drwxr-xr-x      5       0       0       16384   1683382865      .
drwxr-xr-x      4       0       0       4096    1683382865      ..
drwxr-xr-x      3       0       0       4096    1683382865      2
-rw-r--r--      1       0       0       0       0       new_file
drw-r--r--      2       0       0       4096    0       new_dir2
> mv new_file new_file2
> mv new_dir2 new_dir3
> ls -l
drwxr-xr-x      5       0       0       24576   1683382865      .
drwxr-xr-x      4       0       0       4096    1683382865      ..
drwxr-xr-x      3       0       0       4096    1683382865      2
-rw-r--r--      1       0       0       0       0       new_file2
drw-r--r--      2       0       0       4096    0       new_dir3
> chmod 0777 new_dir3
> chmod 0655 new_file2
> ls -l
drwxr-xr-x      5       0       0       24576   1683382865      .
drwxr-xr-x      4       0       0       4096    1683382865      ..
drwxr-xr-x      3       0       0       4096    1683382865      2
-rw-r-xr-x      1       0       0       0       0       new_file2
drwxrwxrwx      2       0       0       4096    0       new_dir3
```

命令的意义没啥好说的。ls、touch、mkdir 这些命令的执行顺序写在 main 函数里了；考虑到这个作业的体量，我没做成交互式 shell。

4、经过我自己修改之后的 img.ext4，可以挂载到系统里。

```
liuweinan@liuweinan-xmu-pc:~/20230505_Storage_system$ sudo umount ./rootfs
liuweinan@liuweinan-xmu-pc:~/20230505_Storage_system$ sudo mount ./img.ext4 rootfs/
liuweinan@liuweinan-xmu-pc:~/20230505_Storage_system$ tree -L 5 rootfs/
rootfs/
├── 1
│   ├── 2
│   │   └── 3
│   │       └── 4
│   │           └── 5
│   ├── new_dir3
│   └── new_file2
└── lost+found  [error opening dir]

7 directories, 1 file
liuweinan@liuweinan-xmu-pc:~/20230505_Storage_system$ sudo umount rootfs
```