

Ext 2 文件系统 中期汇报

黄书鸿 计01 2023-4-16

目录

1. Ext 2 文件系统
2. Arceos 文件系统架构
3. 目前进展
4. 下一周计划

Ext 2 文件系统

与 easy-fs 有什么不同之处？

Block: Ext 2 filesystem **访问磁盘的最小单位**，大小在 1024 bytes 到 4096 bytes

Block group: **磁盘中的块被组织为 block group**，每个 group 中最多包含 $8 * \text{block_size}$ 个块

Directory: **目录使用变长的 entry**，并且用链表的方式组织起来，使得删除条目更加容易

其他功能：访问控制、符号链接、块预分配

Ext 2 文件系统

每个 block group 有自己的 bitmap 和 inode table。在 SuperBlock 后，是 Block group descriptor 表，描述了每个 block group 的信息和使用情况：

```
pub struct BlockGroupDesc {  
    pub bg_block_bitmap: u32, // block bitmap 位置  
    pub bg_inode_bitmap: u32, // inode bitmap 位置  
    pub bg_inode_table: u32, // inode table 位置  
    // 使用情况  
    pub bg_free_blocks_count: u16,  
    pub bg_free_inodes_count: u16,  
    pub bg_used_dirs_count: u16,  
}
```

Ext 2 文件系统

目录中的条目是变长的，支持长度最大 255 的文件名；以链表的方式组织，每个条目记录下一个条目相对于当前的偏移；删除条目时，更改前一个条目的偏移量即可：

```
pub struct DirEntryHead {  
    pub inode: u32,  
    pub rec_len: u16, // 到下一个条目的偏移量  
    pub name_len: u8,  
    pub file_type: u8, // 记录文件类型，不需要额外访问磁盘  
    // 下面是变长的名字  
}
```

Arceos 文件系统架构

- 整体框架支持访问控制;
- 支持将其他的文件系统挂载到主文件系统的目录下;

Arceos 文件系统架构

文件系统的抽象

```
pub trait VfsOps: Send + Sync {  
    /// Do something when the filesystem is mounted.  
    fn mount(&self, _path: &str, _mount_point: VfsNodeRef) -> VfsResult;  
    /// Do something when the filesystem is unmounted.  
    fn umount(&self) -> VfsResult;  
    /// Format the filesystem.  
    fn format(&self) -> VfsResult;  
    /// Get the attributes of the filesystem.  
    fn statfs(&self) -> VfsResult<FileSystemInfo>;  
    /// Get the root directory of the filesystem.  
    fn root_dir(&self) -> VfsNodeRef;  
}
```

Arceos 文件系统架构

VNode 的抽象

```
pub trait VfsNodeOps: Send + Sync {  
    // common operations:  
    fn open(&self) -> VfsResult;  
    fn release(&self) -> VfsResult;  
    fn get_attr(&self) -> VfsResult<VfsNodeAttr>;  
    // file operations:  
    fn read_at(&self, _offset: u64, _buf: &mut [u8]) -> VfsResult<usize>;  
    fn write_at(&self, _offset: u64, _buf: &[u8]) -> VfsResult<usize>;  
    fn fsync(&self) -> VfsResult;  
    fn truncate(&self, _size: u64) -> VfsResult;  
    ...  
}
```


Arceos 文件系统架构

```
// directory operations:
fn parent(&self) -> Option<VfsNodeRef>;
fn lookup(self: Arc<Self>, _path: &str) -> VfsResult<VfsNodeRef>;
fn create(&self, _path: &str, _ty: VfsNodeType) -> VfsResult;
fn remove(&self, _path: &str) -> VfsResult;
fn read_dir(&self, _start_idx: usize, _dirents: &mut [VfsDirEntry]) -> VfsResult<usize>;
} // VfsNodeOps
```

Arceos 文件系统架构

具体实现

```
/home/huangshuhong/OS/arceos/modules/axfs
```

```
├── Cargo.toml
├── src
│   ├── api (4)
│   │   ├── dir.rs // 实现了读取、创建目录的 api
│   │   ├── file.rs // 实现了创建、打开、读写文件的 api
│   │   └── mod.rs
│   ├── dev.rs // 封装了 axdriver::VirtIoBlockDev (0)
│   ├── fops.rs // 进一步封装 VfsNodeOps 为 Dir 和 File, 使贴近 Linux 的规范 (3)
│   ├── fs
│   │   ├── fatfs.rs // 基于 fatfs 实现了 VfsNodeOps 和 VfsOps trait (1)
│   │   └── mod.rs
│   ├── lib.rs
│   ├── root.rs // 实现了根目录、其他文件系统也可以挂载 (2)
├── tests
│   └── test_axfs.rs
```

目前进展

在 `crates/ext2fs` 中仿照 `easy-fs` 写了文件系统的接口，目前支持：创建ext2文件系统镜像、从镜像中打开文件系统 `create_file`、`create_dir`、`link`、`unlink` 等功能。

```
// create_fs, open_fs, create_file, ls --- crates/ext2fs_fuse/src/main.rs
Ext2FileSystem::create(block_file.clone());
let efs = Ext2FileSystem::open(
    block_file.clone(),
    SystemTime::now().duration_since(UNIX_EPOCH).unwrap().as_secs() as u32);
let root_inode = Ext2FileSystem::root_inode(&efs);
root_inode.create("filea", EXT2_S_IFREG);
root_inode.create("fileb", EXT2_S_IFREG);
```

目前进展

```
// unlink、mkdir --- crates/ext2fs_fuse/src/main.rs
root_inode.unlink("fileb");
root_inode.create("dir_a", EXT2_S_IFDIR);
let dir_a = root_inode.find("dir_a").unwrap();
dir_a.create("filec", EXT2_S_IFREG);
let filec = dir_a.find("filec").unwrap();
filec.write_at(0, greet_str.as_bytes());
let len = filec.read_at(0, &mut buffer);
assert_eq!(greet_str, core::str::from_utf8(&buffer[..len]).unwrap());
```

目前进展

```
$ RUST_BACKTRACE=1 cargo run --package ext2fs_fuse --bin ext2fs_fuse
```

After create filea and fileb:

.

..

filea

fileb

After **unlink**:

.

..

filea

Under dir_a:

.

..

filec

下一周计划

1. 将 ext2 集成到目前的 Arceos 的文件系统框架中；
2. 进一步完善 ext2 文件系统的功能，比如支持软链接、unlink 一个目录（目前只支持文件）、文件状态；
3. 调研带日志的文件系统的实现，准备把 ext2 升级至 ext3。

欢迎提问