

Project 6 File System 设计文档

中国科学院大学

吴俊亮

2021 年 1 月 11 日

1. 文件系统初始化设计

磁盘的布局如下图所示，前 2048 个扇区用于内核镜像，第 2048-4095 个扇区用于交换内存，第 4096-2102271 个扇区用于文件系统。super block 使用第 4096 个扇区；block map 从第 4097 个扇区开始，占用 64 个扇区；inode map 使用第 4161 个扇区；inode table 从第 4162 个扇区开始，占用 512 个扇区；数据区从第 5120 个扇区开始，占用 2097152 个扇区，共计 1GB。

```

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Disk Layout
|      0 -      2,047 (      2,048): Kernel Image
2,048 -      4,095 (      2,048): Swap

4,096 -      4,096 (          1): Super Block
4,097 -      4,160 (          64): Block Map
4,161 -      4,161 (          1): Inode Map
4,162 -      4,673 (         512): Inode
5,120 - 2,102,271 (2,097,152): Block
* * * * * * * * * * * * * * * * * * * * * * * * * * * */

```

super block 和 inode 的数据结构如下图所示。super block 中记录了 super block、block map、inode map、inode 和数据区的起始扇区和占用的扇区数，根目录的 inode 和数据所在的位置以及目前已使用的 inode 和 block 的数量。inode 中记录了文件的类型，拥有者，大小以及数据块的位置（包括 10 个直接指针和 2 个一级间接指针）。该文件系统能支持的最大文件约为 8MB，最多支持 4096 个文件或目录，单个目录下最多可有 128 个文件或子目录。

```
// 512 bytes
typedef struct superblock{
    uint64_t magic;
    uint64_t superblock_id;
    uint64_t superblock_num;
    uint64_t blockmap_id;
    uint64_t blockmap_num;
    uint64_t inodemap_id;
    uint64_t inodemap_num;
    uint64_t inode_id;
    uint64_t inode_num;
    uint64_t block_id;
    uint64_t block_num;
    uint64_t root_inode_id;
    uint64_t root_block_id;
    uint64_t used_inode_num;
    uint64_t used_block_num;
    uint8_t empty[392];
} superblock_t;

// 64 bytes
typedef struct inode{
    uint32_t mode;
    uint32_t owner;
    uint32_t size;
    uint32_t direct_blocks[10];
    uint32_t indirect_blocks[2];
    uint32_t empty;
} inode_t;
```

2. 文件操作设计

创建一个文件需要申请一个空的 inode，并申请一个上级目录的目录项，将该目录项指向新的 inode。新的 inode 需要初始化其 mode、owner、size。打开一个文件需要申请一个文件描述，并将文件描述的下标返回给用户程序。文件描述的数据结构如右图所示，包括 inode 的序号，访问权限和读写指针。

```
typedef struct fd{
    uint16_t inode;
    uint16_t access;
    uint64_t rd_pos;
    uint64_t wr_pos;
} fd_t;
extern fd_t fd_list[10];
```

3. 目录操作设计

以 ls 命令为例，路径查找的过程大致如下：

- ① 检查第一个字符是否为‘/’，是则从根目录开始查找，否则从当前目录开始查找。
- ② 得到下一级目录的名称，在当前目录中查找下一级目录。
- ③ 判断是否为最后一级目录，是则结束查找，否则跳②继续查找。

该文件系统支持多级目录的查找。