

Project 6 File System 设计文档

中国科学院大学

蔡润泽

2019 年 12 月 29 日

1. 文件系统初始化设计

- (1) 请阐述你设计的文件系统对磁盘的布局（可以使用图例表示），包括从磁盘哪个位置开始，superblock, inode map, block/sector map, inode table 以及数据区各自占用的磁盘空间大小

在本设计中，磁盘的布局图如下：

Super Block	Block Map	Inode Map	Inode	Data Block
-------------	-----------	-----------	-------	------------

其中 FS 的磁盘偏移量为 512MB。superblock 占用 512B, Block map 占用 16KB, inode map 占用 512B, inode table 占用 256KB, 数据区占用 511.7MB。

- (2) 请列出你设计的 superblock 和 inode 数据结构，并阐明各项含义。请说明你设计的文件系统能支持的最大文件大小，最多文件数目，以及单个目录下能支持的最多文件/子目录数目。

Superblock 的数据结构如下：

```
//Superblock
typedef struct superblock{
    uint32_t magic_num;
    uint32_t fs_sz;
    uint32_t start_sector;

    uint32_t blockmap_addr;
    uint32_t blockmap_offset;
    uint32_t blockmap_num;

    uint32_t inodemap_addr;
    uint32_t inodemap_offset;
    uint32_t inodemap_num;

    uint32_t inodes_addr;
    uint32_t inodes_offset;
    uint32_t inodes_num;

    uint32_t datablock_addr;
    uint32_t datablock_offset;
    uint32_t datablock_num;

    uint32_t inode_sz;
    uint32_t dentry_sz;
} superblock_t;
```

包括了 Magic Number, FS 的大小, 起始的扇区, 以及第一问中设计的所有布局的磁盘起始地址, 相较于起始扇区的偏移量, 以及占用的扇区数。Superblock 中还记

录了 inode 和 dentry 的大小。

inode 的数据结构如下：

```
#define INODE_SIZE      64
typedef struct inode{
    uint32_t inum;
    file_mode_t mode;           //r/w
    uint16_t i_mode;            //0:file 1:dentry
    uint16_t ref;               //link num
    uint32_t used_sz;           //used size in data block
    uint32_t ctime;             //created time
    uint32_t mtime;             //modified time
    uint32_t num;               //used num of data block or dentry
    uint32_t direct[MAX_DIR_BLK]; //direct ptr to data block
} inode_t;
```

inum 表示 inode 的序号，mode 表示 inode 所指的类型读写的限制，i_mode 表示 inode 所指的类型是文件还是目录项，ref 表示链接数，used_sz 表示 inode 所指的 data block 所使用的字节大小。ctime 表示创立时间，mtime 表示修改时间。num 表示 inode 所使用的 data block 或者目录项的个数。direct 表示 9 个直接指向的 data block 块。

本设计中，支持的最大文件大小为 36KB，最多文件数目为 4096 个，以及单个目录下能支持的最多文件/子目录数目不超过 4096 个（4095）。

- (3) 设计或实现过程中遇到的问题和得到的经验（如果有的话可以写下来，不是必需项）

在 QUME 上运行代码时，发现 sdread 和 sdwrite 函数报错，经过排查发现时 QUME 分配的磁盘过小，后面重新分配了一个 512MB 大小的虚拟磁盘空间，该问题得到解决。

2. 文件操作设计

- (1) 请说明创建一个文件所涉及的元数据新增和修改操作，例如需要新增哪些元数据，需要修改哪些元数据

创建一个文件需要修改指向该文件 inode 的 inum, mode, i_mode, ref, used_sz, ctime, mtime, num 以及 datablock 的地址。同时还需要对所在目录的 inode 进行修改，需要修改目录节点 ref, num 以及 used_sz。

3. 目录操作设计

- (1) 请说明文件系统执行 ls 命令查看一个绝对路径时的操作流程

文件系统查找绝对路径需要进行递归查找。首先从根节点开始查找，找到根节点的目录项表内有没有绝对路径下第一个子目录的路径，若有，则继续查找。依次递归查找，知道找到最后一个子目录，此时 ls 将该目录所存的文件名依次打印输出即可。

参考文献

- [1] XV6 开发文档
- [2] project_6_guid_book