Week14 Report in Class (Fri56)

11911839 聂雨荷

Q1

手册第二部分中的 inode，如果 block_size 是4KB，指针大小是4B，一个 inode 能管理的最大文件大小是多少，写出计算过程。

- 直接指针：12 x 4KB = 48 KB
- 一级间接指针
  - 一个 block 可以存放：4KB/4B = 2^{10} 个指针
  - 所以 2^{10} 个 block 可以存放 2^{10} x 4KB = 4MB 大小的文件
- 二级间接指针
  - 一个 block 可以存放：4KB/4B = 2^{10} 个指针
  - 两层可以存在 2^{20} 个指针
  - 所以 2^{20} 个 block 可以存放 2^{20} x 4KB = 4GB 大小的文件
- 三级间接指针
  - 一个 block 可以存放：4KB/4B = 2^{10} 个指针
  - 三层可以存在 230^ 个指针
  - 所以 230^ 个 block 可以存放 2^{30} x 4KB = 4TB 大小的文件

所以一个 inode 能够管理的最大文件大小为 48KB + 4MB + 4GB + 4TB

Q2

SFS 中的 inode 可以管理的最大文件大小是多少，写出计算过程。

SFS 中的 innode 指向的 block_size 是 4KB，指针为 4B，与 Q1 的计算方法相同

SFS 中存在直接指针和一级间接指针

```
uint32_t direct[SFS_NDIRECT]; //此inode的直接数据块索引值（有SFS_NDIRECT个）
uint32_t indirect; //此inode的一级间接数据块索引值
```

故如 Q1 计算方式，SFS 中的 inode 可以管理的最大文件大小是 48KB + 4MB

Q3

SFS 中 sfs_disk_inode 和 sfs_disk_entry 的关系是什么。

```
/*inode (on disk)*/
struct sfs_disk_inode {
    uint32_t size; //如果inode表示常规文件，则size是文件大小
    uint16_t type; //inode的文件类型
    uint16_t nlinks; //此inode的硬链接数
    uint32_t blocks; //此inode的数据块数的个数
    uint32_t direct[SFS_NDIRECT]; //此inode的直接数据块索引值
 (有SFS_NDIRECT个)
    uint32_t indirect; //此inode的一级间接数据块索引值
};
```

```
/* file entry (on disk) */
struct sfs_disk_entry {
    uint32_t ino; //对应文件的inode索引
    char name[SFS_MAX_FNAME_LEN + 1]; //文件名
};
```

SFS 中的 sfs_disk_inode 代表了一个实际位于磁盘上的文件。

- 对于普通文件，sfs_disk_inode 中的索引值指向的 block 中保存的是文件中的数据
- 对于目录，sfs_disk_inode 索引值指向的数据保存的是目录下的 sfs_disk_entry

SFS 中的 sfs_disk_entry 数据结构中，name 表示目录下文件或文件夹的名称，ino 表示磁盘 block 编号，通过 读取该 block 的数据，能够得到相应的文件或文件夹的 inode。ino 为 0 时，表示一个无效的 entry。

- 每个 sfs_disk_entry 占一个 Block 文件项内存储了指向文件 inode 的索引值以及文件名