

FS Lab

基础结构的设定

采用VSFS

题目要求：65536个块，最多32768个文件，单个文件最多8M，一个块大小4K，路径没有.和..

原本的VSFS里super block需要占用一个块，但是考虑到我们直接从ROOT=0的根节点开始建立文件树，并且文件系统本身的信息可以不放入块内，因此我没有再将super block的信息存入，而是查询时直接修改

计算一个块最多记录 $4 \times 1024 \times 8 = 32768$ bit信息，刚好对应32768个文件，因此inode bitmap只需要占用一个块

同理data bitmap需要占用两个块，然后分配sizeof(inode)和inode占用的块

观察题目中已有的stat结构，mode可以简化成bool，同时n_link等有些信息基本上用不到（测试数据里只有简单的建立文件，目录和删除文件，目录的指令）。因此只保留有用的atime, mtime, ctime, size和bool类型的mode信息，此时已有29字节(size_t8字节一个，放在最前面，size是int类型，4字节，bool类型1字节放最后)

因此我们考虑将inode的大小补齐在32字节，根据对齐规则，只能使用到3个char类型或者1个short类型的变量用来保存inode指向的块信息。

因此原本的指针形式的保存方式不再适用，直接采用保留块编号的信息（因为测试数据中的单个文件大小都很少，基本都只有几字节）。但是如果真的需要保存大文件，单个文件最大8M，因此需要 $8M/4K = 2048$ 个块，还是应该采用2个indirect指针，这里为了方便完成直接保留块编号了。

同时考虑到，最大块数量是65536，为2的16次方，若采用short无法保存，应为unsigned int，即uint16_t，刚好保存下来(虽然用不到这么多)

综上，datablock-bitmap从0开始编号，inode-bitmap从2开始编号，inode-block从3开始编号

为了方便可以直接定义宏来表示各个部分的起始位置，同理，若此后对inode结构有所调节的话也会更为方便

同时记录全局变量if_error，用来保存-ENOENT的信息

结构完善

结构完善是参考了网上的代码，其中有很多方法写的很好，比如将常用功能抽象成函数等

具体结构

```
typedef uint16_t block_id_t;
typedef uint16_t inode_id_t;
//文件结构
typedef struct {
    time_t atime;
    time_t mtime;
    time_t ctime;
    int size;
    bool mode;
    block_id_t ptr_block;
} inode_t;
typedef char block_t[BLOCK_SIZE];
//所有的bitmap直接当数组用而不是指针
typedef uint64_t inode_bitmap_t[1*BLOCK_SIZE/sizeof(uint64_t)];
typedef uint64_t block_bitmap_t[2*BLOCK_SIZE/sizeof(uint64_t)];
//目录结构
typedef struct {
    char file_name[MAXFILENAME];
    inode_id_t inode_id;
}directory;
//超级块结构
typedef struct {
    unsigned long bsize;
    fsblkcnt_t blocks;
    fsblkcnt_t bfree;
    fsblkcnt_t bavail;
    fsfilcnt_t files;
    fsfilcnt_t ffree;
    fsfilcnt_t favail;
    unsigned long namemax;
}vfs_t;
```

1.关键的核心代码

```
void locate_inode(inode_id_t inode_id, block_id_t *block_id, int *offset)
```

用来找inode的块编号及偏移量

```
void fill_inode(inode_id_t inode_id, bool flag)
```

填充块

```
void move_inode(inode_id_t inode_id)
```

移动块，用于文件操作的目录删除和硬链接删除（不同于删除块），删除链接并未删除原文件

```
inode_t find_path(const char* path, inode_id_t* inode_id)
```

用于使用路径查找inode编号

```
inode_t find_parent_path(const char *path, inode_id_t  
*dir_inode_id, char buff[MAXFILENAME+1], bool *exists)
```

用于使用了路径查找父节点inode编号

其中fs_read, fs_write, fs_truncate基本都是参照的网上代码，自己实在写不来

尤其是在完成read, write和truncate功能时遇到了很多问题，一开始总是找不到文件位置，然后是修改没有写到文件里，debug也调不出来，难度较大，解决方案是参照网上代码

2. 注意事项

每次修改完inode后，记得修改对应的inode-block

每次删除或新建block，记得修改对应的datablock-bitmap

目录结构是较为特殊的文件，只有目录名信息和对应的inode编号

在文件读写时，记得都需要修改访问时间，都需要修改inode，如果是追加写记得更新文件size（测试数据好像没有追加写）

测试结果展示

总体功能完成，下述为traces中的测试

0.sh:

```
[2020200671@work122 fslab-handout]$ ./traces/0.sh
```

1.sh

```
[2020200671@work122 fslab-handout]$ ./traces/1.sh  
dir1 dir2  
dir3
```

2.sh

```
[2020200671@work122 fslab-handout]$ ./traces/2.sh
file1 file2
abcdefghijklmnopqrstuvw file1 file2
```

3.sh

```
[2020200671@work122 fslab-handout]$ ./traces/3.sh
0
32
10000
24
```

4.sh

```
[2020200671@work122 fslab-handout]$ ./traces/4.sh
2012-12-21 00:00:36.000000000 -05002012-12-21 00:00:36.000000000 -0500
2012-12-21 00:00:36.000000000 -05002012-12-21 00:00:36.000000000 -0500
```

5.sh

```
[2020200671@work122 fslab-handout]$ ./traces/5.sh
dir1 dir3
dir3
dir1 dir3
```

6.sh

```
[2020200671@work122 fslab-handout]$ ./traces/6.sh
file1 file2 picture
file1 file2
file1
```

7.sh

```
[2020200671@work122 fslab-handout]$ ./traces/7.sh
dir2 file1
dir2 file2
dir3 file2
dir
dir3 file2
```

8.sh

```
[2020200671@work122 fslab-handout]$ ./traces/8.sh
dir1-1 dir1-2
dir1 dir2
dir2
dir2-1
dir1-1 dir1-2
dir1-1 dir2
dir1-2
```

9.sh

```
[2020200671@work122 fslab-handout]$ ./traces/9.sh
```

10.sh

```
[2020200671@work122 fslab-handout]$ ./traces/10.sh
12345
23333333333
23333333333
abcdefg
```

11.sh

```
[2020200671@work122 fslab-handout]$ ./traces/11.sh
```

12.sh

```
[2020200671@work122 fslab-handout]$ ./traces/12.sh
File: 'file1'
Size: 0          Blocks: 0          IO Block: 4096   regular empty file
Access: (0644/-rw-r--r--) Uid: ( 1010/2020200671)   Gid: ( 1003/2021ICS_1)
File: 'file1'
Size: 6          Blocks: 8          IO Block: 4096   regular file
Access: (0644/-rw-r--r--) Uid: ( 1010/2020200671)   Gid: ( 1003/2021ICS_1)
```

13.sh

```
[2020200671@work122 fslab-handout]$ ./traces/13.sh
```

14.sh

```
[2020200671@work122 fslab-handout]$ ./traces/14.sh
dir0   dir20   dir323  dir447  dir570  dir694  dir817  dir940  file141  file265  file389  file511  file635  file759  file88
dir1   dir200  dir324  dir448  dir571  dir695  dir818  dir941  file142  file266  file39  file512  file636  file76  file88
dir10  dir201  dir325  dir449  dir572  dir696  dir819  dir942  file143  file267  file390  file513  file637  file760  file88
dir100  dir202  dir326  dir45  dir573  dir697  dir82  dir943  file144  file268  file391  file514  file638  file761  file88
dir1000  dir203  dir327  dir450  dir574  dir698  dir820  dir944  file145  file269  file392  file515  file639  file762  file88
dir1001  dir204  dir328  dir451  dir575  dir699  dir821  dir945  file146  file27  file393  file516  file64  file763  file88
dir1002  dir205  dir329  dir452  dir576  dir7  dir822  dir946  file147  file270  file394  file517  file640  file764  file88
dir1003  dir206  dir33  dir453  dir577  dir70  dir823  dir947  file148  file271  file395  file518  file641  file765  file88
dir1004  dir207  dir330  dir454  dir578  dir700  dir824  dir948  file149  file272  file396  file519  file642  file766  file89
dir1005  dir208  dir331  dir455  dir579  dir701  dir825  dir949  file15  file273  file397  file52  file643  file767  file89
dir1006  dir209  dir332  dir456  dir58  dir702  dir826  dir95  file150  file274  file398  file520  file644  file768  file89
dir1007  dir21  dir333  dir457  dir580  dir703  dir827  dir950  file151  file275  file399  file521  file645  file769  file89
dir1008  dir210  dir334  dir458  dir581  dir704  dir828  dir951  file152  file276  file4  file522  file646  file77  file89
dir1009  dir211  dir335  dir459  dir582  dir705  dir829  dir952  file153  file277  file40  file523  file647  file770  file89
dir101  dir212  dir336  dir46  dir583  dir706  dir83  dir953  file154  file278  file400  file524  file648  file771  file89
dir1010  dir213  dir337  dir460  dir584  dir707  dir830  dir954  file155  file279  file401  file525  file649  file772  file89
```

15.sh

```
[2020200671@work122 fslab-handout]$ ./traces/15.sh
```