

# 任务二：文件系统 API 实现

中国科学院大学 操作系统研讨课

2016.12.14

## 1. 任务要求

实现一个简单的文件系统：

- 实现任务书中要求的所有文件系统接口
- 至少支持间接索引
- 不需要考虑文件系统一致性保障(如 journal、COW 等)
- 支持绝对路径与相对路径解析

文件系统需要通过给定测试用例，并自行设计其他测试用例。

## 2. Start-code 介绍

### 2.1. 文件介绍

- block.h: 提供块设备接口，文件系统通过这些接口访问块设备(block device)。
- blockFake.c: 在当前目录创建一个名为 disk 的文件，来模拟一个块设备。  
上述文件请不要修改
- shell.c: 提供部分 shell 命令的实现，需要实现 ls 命令。
- fs.h, fs.c: 文件系统接口，需要实现所有的接口。
- test.py: 测试用例 python 脚本。

### 2.2. 获取

课程网站。

### 2.3. 运行

Makefile 文件提供编译功能。

make 编译命令。

./test.py 或 python test.py 运行。

## 3. 任务

### 3.1. 设计和评审

帮助学生发现设计的错误，及时完成任务。学生需要对这次的作业进行全面考虑，在实现代码之前有清晰的思路。学生讲解设计思路时可以用不同的形式，如伪代码、流程图等，建议使用 PPT。

### 3.1.1. 设计介绍

- 文件系统初始化流程，如何实现 mkfs？
- 打开文件及关闭文件的流程。
- 读写文件的流程，改变读写位置(lseek)接口的实现。
- 创建与删除目录的流程。如何实现 ls？
- 硬链接 link 与 unlink 的实现。
- 文件状态信息打印(fs\_stat)。

## 3.2. 开发

### 3.2.1. 要求

- 实现 fs.h 中定义的所有接口。
- 至少在磁盘上保存 2 份 superblock。
- 实现 shell.c 中 ls 命令，ls 需支持绝对路径和相对路径，如 “ls /x/y/”。
- 通过提供的测试用例 test.py，并根据自己的文件系统实现，提供 3 个边界条件测试用例。

### 3.2.2. 注意事项

- 磁盘大小 FS\_SIZE 在 fs.h 中定义，单位为扇区数(sectors)。假设操作系统中只有一个磁盘，并可以根据需要改变 FS\_SIZE 来调整磁盘的大小。
- **不要求**内核可启动，但模拟器**必须**实现。fs.c 中 #ifdef FAKE, Makafile 的 lxns 中默认传入 -DFAKE, 即默认编译成文件系统模拟器。模拟器不依赖于原有 kernel, 如果要使文件系统可以在之前的内核中启动起来，需要注意 fs 接口实现中用 ifdef 区分不同部分。鼓励实现可启动内核的文件系统。
- 多份 superblock 同步修改。当其中一份 superblock 发生损坏时，mkfs 依然能通过读取未损坏的 superblock 来识别出已格式化的文件系统。
- rmdir 接口需要支持递归删除给定目录及其子目录下所有内容。
- shell 的 ls 命令需要扩充参数，支持带路径参数的 ls，例如 ls /home/test/
- 自己编写的测试用例需要根据文件系统元数据的设计，测试边界条件，例如单个文件最大 size 测试，可以分别创建小于最大 size，等于最大 size，超过最大 size 的文件进行测试。
- 文件系统相关 shell 命令介绍，见下表。

| 命令    | 参数                | 描述  |
|-------|-------------------|---|
| mkfs  |                   | 创建新的文件系统  |
| open  | <filename> <flag> | 用给定标记打开文件，返回响应的文件描述符<br><flag>: 1: FS_O_RDONLY; 2: FS_O_WRONLY; 3: FS_O_RDWR<br>注意：当文件打开时相应的偏移量为 0。 |
| read  | <fd> <size>       | 从文件中读取 size 个字节，同时文件偏移量向前移动 size 个字节。   |
| write | <fd> <string>     | 向文件的当前偏移量处写入字符串，同时文件偏移量移动相应字节。  |
| lseek | <fd> <offset>     | 改变文件偏移量位置   |
| close | <fd>              | 关闭文件  |

|        |                    |  |
|--------|--------------------|--|
| mkdir  | <dirname>          | 在当前目录下创建子目录                              |
| rmdir  | <dirname>          | 删除目录                                     |
| cd     | <dirname>          | 改变当前工作目录                                 |
| link   | <src> <dest>       | 创建硬链接                                    |
| unlink | <name>             | 删除链接                                     |
| stat   | <name>             | 显示文件或目录的状态。包括 inode 信息，类型，链接数，大小，已分配的块数等 |
| ls     | <optional:dirname> | 显示当前目录下的内容(文件与子目录名)<br><b>需要实现该功能</b>    |
| cat    | <filename>         | 显示文件内容                                   |
| create | <filename><size>   | 在当前目录下创建文件                               |

## 4. 任务测试

### 4.1. 测试方法

```
bootblock.o floppy.img kernel mb
bootblock.s fs.c kernel.c mb
→ project6 ./test.py
```

### 4.2. 测试结果

部分测试结果如下图所示，其余特殊情况参加 test.py 中的注释。

```
.....Starting my tests

*****Make File System Test*****
ShellShock Version 0.000003

# # # # Name                                Type Inode Size
.                                           D    0    320
..                                          D    0    320
f1                                           F    1    10
f2                                           F    2    10
f3                                           F    3    10
# # Name                                Type Inode Size
.                                           D    0    128
..                                          D    0    128
# Goodbye

*****
*****Simple Errors Test*****
ShellShock Version 0.000003

# # Problem with removing directory
# Problem with changing directory
# Problem with link
# Problem with unlink
# Goodbye

*****
*****Open Test*****
ShellShock Version 0.000003

# # # File handle is : 0
# Data read in : AB
# Error while writing file
# OK
# ABCDEFGHIJ
# Goodbye
```

### 4.3. py 测试脚本编写

修改 test.py 即可。

def 定义测试名称函数，在函数中用 issue 发送 shell 命令。并在 py 文件末尾添加对测试的调用。

示例:

```
def my_test ():  
    issue('shell cmds')  
    do_exit()
```

Python For 循环示例:

```
for x in range(1, 121):  
    issue('create f%d 1' %x)
```

## 参考资料

[1] [\[单击此处键入参考文献内容\]](#)

