

# OSLab 5 文件系统 实验报告

姓名：凌嘉伟 学号：151220061

邮箱：151220061@smail.nju.edu.cn

## 一 实验进度

- 1、总体实现了一个文件系统，可以直接以文件名的形式读取文件（由于能力等不可抗力因素，原游戏文件未能读入，只能写了两个念诗（“苟…”）的文件进行读写操作）
- 2、用 format 格式化磁盘并且生成了镜像文件 disk.img
- 3、实现了文件操作的系统调用（完成了 open,close,read,write,lseek，另外加上 rewind 函数将文件指针重新指向文件开头（其实就是 lseek 操作））
- 4、实现了 format 磁盘格式化工具（由于能力、时间等不可抗力因素，文件直接采取调用库函数的形式读取文件，而并没有写 copy2myfs 和 read\_myfs）

## 二 实验环境

Ubuntu 16.10

## 三 实验说明

- 1、为了测试文件系统，在游戏中写了一个念诗的程序 fs\_test，并在 makefile 中将接口换为 fs\_test，建立两个文件 readpoem.txt 和 writepoem.txt（其实就是某两句诗），在 fs\_test 中简单的通过文件的系统调用来进行文件操作。

```
int fs_test()
{
    char poem[10][30];
    int i;
    printf("Let us read poem and continue 1 second!\n");
    int fp1 = file_open("readpoem.txt", 0);
    printf("fp1 = %d\n", fp1);
    for (i = 0; i < 4; i++)
    {
        file_read(fp1, (void *)poem[i], 30);
        printf("%s\n", poem[i]);
    }
    file_close(fp1);
    int fp2 = file_open("writepoem.txt", 0);
    printf("fp2 = %d\n", fp2);
    for (i = 0; i < 4; i++)
    {
        file_write(fp2, (void *)poem[i], 30);
        printf("%s\n", poem[i]);
    }
    memset(poem, 0, sizeof(poem));
    file_rewind(fp2);
    for (i = 0; i < 4; i++)
    {
        file_read(fp2, (void *)poem[i], 30);
        printf("%s\n", poem[i]);
    }
    file_close(fp2);
    printf("The poem has been read and written and we have continue 1s for him 0_0!\n");
    while (1);
    return 0;
}
```

- 2、完成了 open,close,read,write,lseek 的操作（其实 PA 里面已经写过了），文件在 kernel/fs/fs.c 中
- 3、实现了一个 format.c 的磁盘格式化文件，生成了 disk.img 的镜像，由于时间和能力问题，

并没有写其他两个 `copy2myfs` 和 `read_myfs`

#### 四 实验问题

- 1、在实验过程中，发现 `format.c` 加载文件时，将游戏文件加载后，参数 `argc` 的值竟然会变，因此调试了很久，玄学般调出来了……
- 2、在实验过程中，发现页表初始化出现了问题，`page_free_list` 压在内核尾部，其实运行过程中，这些页有内容，`page_alloc` 分配了一个非空的页表，因此将初始化页表的范围调整了

#### 五 实验感想

终于将 OSLab 赶到这个地方了，其实促进的动力还是什么都不太会，但并不想挂掉这门课 TAT（只求及格是不是要求太(现)低(实)了点……）。奈何自己能力问题，很多地方并没有透彻掌握，因此 OSLab 写的很乱很糟糕 QAQ，尤其是从 Lab3 开始，总是大改改了很多，bug 也调了很久，一度想放弃这个。然而为了实验的完成度，还是忍着往前赶，最后只求能拿到分就行，最后三次 Lab，第 3 次有 bug 然而未能改过来，第 4 次投机取巧用了具名信号量，最后一次为了尽可能在能力范围内拿到分数只能部分完成要求。总之在 OSLab 中，我还是学到很多的，尤其是那种不到ddl不死心（划掉），不能放弃。同时也感谢同学以及助教帮助，有时候一次不经意的 debug 就能扭转反杀：)