

## oslab5 文件系统 实验报告

141242026 刘驭壬

邮箱: [141242026@smail.nju.edu.cn](mailto:141242026@smail.nju.edu.cn)

### 实验进度:

实现了一个简易的文件系统，可以在游戏中以文件名的形式读取文件。

具体的有以下方面：

1. 实现格式化程序
2. 实现了 open,close,read,write,lseek,rewind 等文件操作函数
3. 重新改写了 kernel 的 loader，使用文件操作函数读
4. 编写了文件操作函数的测试程序并测试通过

### 关于实验的部分说明:

- 1.对于格式化程序，由于基本按照助教 ppt 上的说明写，未遇到很大问题。
- 2.open,close,read,write,lseek,rewind 等文件操作函数仿照了 ics2015 的相关函数，做了一定修改。
- 3.对于测试正确性。Make qemu 后会运行 game.c 中的 fs\_test 函数。该函数验证了实现的文件操作函数。

```
00
01
02 int fs_test(){
03     printg("game start\n");
04     int fp1=open("poem1.txt",0);
05     printg("fp1=%d\n",fp1);
06     char poem[10][30];
07     int i;
08     for(i=0;i<4;i++){
09         read(fp1,(void*)poem[i],30);
10         printg("%s\n",poem[i]);
11     }
12     close(fp1);
13     int fp2=open("poem2.txt",0);
14     printg("fp2\n");
15     for(i=0;i<4;i++){
16         write(fp2,(void*)poem[i],30);
17         printg("%s\n",poem[i]);
18     }
19     memset(poem,0,sizeof(poem));
20     frewind(fp2);
21     for(i=0;i<4;i++){
22         read(fp2,(void*)poem[i],30);
23         printg("%s\n",poem[i]);
24     }
25     close(fp2);
26     printg("game over nice job");
27     while(1);
28     return 0;
29 }
```

### 出现的问题:

```
region_alloc(penv,(void*)(USTACKTOP-1024*PGSIZE),1024*PGSIZE);  
penv->env_tf.eip=elf->entry;
```

在重写 loader 的时候，我分配完堆栈以后一直缺页，经过多次 print 查看发现，elf->entry 在 region\_alloc 之后被改写了，我对此查了很久，由于之前有关分配页的函数没有发生过这种问题，很难想到有什么地方会改写它。所以我几乎是每个函数跟踪看看在哪个地方改写的。最后查到是在一次 page\_alloc 的时候将页的内容清零了。我就想为什么会 page\_alloc 有内容的页呢？查了很久 page\_alloc 的代码也让很多同学帮忙看，还是没发现错误。结果最后查出来原因是

```
while(1);  
unsigned long i;  
for (i = npages-1; i>0x200; i--) {  
    pages[i].pp_ref = 0;  
    pages[i].pp_link = page_free_list;  
    page_free_list = &pages[i];  
}
```

当初初始化页的时候我压的很死，page\_free\_list 刚好压在了 kernel 尾部的地方，后来 kernel 长了，这些页其实已经有内容了，所以 page\_alloc 实际分配了一个非空的页，后来改成 i>0x200，错误修复。（巨坑，bug 比较深，查了很久）