

Lab 5

- lyy0627@sjtu.edu.cn, 李逸岩, 519021911103

1 tmpFS

最主要处理的Bug 是一个局部变量的指针使用。在分配node的时候未使用给定的Helper `new_dent`, 导致 `tfs_creat` 之后会出现仍然找不到文件的情况。

此外, `tfs_file_write` 也花费了一些时间。原来认为需要讨论诸多情况, 包括原文件边界(左右), 新写入内容边界(左右)以及PAGE SIZE边界。光是新Content和原文件边界讨论就需要6种情况。但后来发现只需要简单的检查“当前页”+“当前页内偏移量”, 获取/创建新页, 然后更新文件大小即可。重构了代码。

2 Shell

解决了一个Lab 4残留的Bug, 发现在 `connect_fs` 时有VMSpace重复映射的问题。找到是建立连接时发生了重复映射。因为无论是Source还是Target, 在4的实现中都同时映射了client/server的Buffer, 究其原因是测试时会发生Page Fault. Server的Buffer会被Client的虚拟地址访问, 于是当时决定直接映射(这是不符合语义的)。

再究其原因则是IPC Call中的地址在传递时没有转换, 正确的做法是:

```
arg = (u64)ipc_msg - conn->buf.client_user_addr + conn-  
>buf.server_user_addr;
```

而不是直接 `(u64)ipc_msg`. 这个Bug解决后, Shell剩下的部分主要需要读FS部分的IPC处理代码, 理解数据存在 `ipc_msg` 的data段, 储存格式是 `dentry` 数组。

3 VFS

解决了Lab 4残留的Bug. 发现在Secondary Start中多次 `init_sched`. 导致多次调用 `rr_sched_init`. 但是本身 `sched_init` 就为每个核准备好了一个IDLE线程并且加入了 `idle_cap_group` 的thread list. 导致可能会在进行Lab 4的测试时, 有几率出现内存错误。

然后我们需要支持所有tmpfs支持的操作，基于操作类型+指明文件方式选择Switch case. 要注意Open时应当将注册的Client核对应的badget和fd绑定，以便下次直接使用Client和FS通信。

随后重构了代码，抽离出sent ipc by name/fd两个函数。注意ipc_call返回值。由于ls等命令依靠这个返回值指示返回长度，应当把ret置为ipc_call的返回值。

在完成上面三个部分后，连续测试30次均能拿到所有分数。

```
lee@Lees-desktop:~/OS/chcore-lab$ make grade LAB=5
=====
Grading lab 5...(may take 20 seconds)
GRADE: tmpfs: tfs_load_image : 15
GRADE: tmpfs: mkdir : 5
GRADE: tmpfs: read_write : 5
GRADE: tmpfs: unlink : 5
GRADE: shell: echo : 3
GRADE: shell: ls : 8
GRADE: shell: cat : 8
GRADE: shell: readline : 4
GRADE: shell: do_complement : 9
GRADE: shell: run cmd : 3
GRADE: fs: fread & fwrite : 7
GRADE: fs: fprintf & fscanf : 8
GRADE: fs: FSM getdents : 8
GRADE: fs: FSM read file : 6
GRADE: fs: FSM optimization : 6
=====
Score: 100/100
```