

## Ex0

按照之前的方式 diff/patch，运行报错

```
trap: setup timer interrupts
not valid addr 4, and can not find it in vma
trapframe at 0xc0337d70
edi 0x00000000
esi 0xc0337f54
ebp 0xc0337dec
oesp 0xc0337d90
ebx 0xc010d3fd
edx 0xc0337e34
ecx 0x00000000
eax 0x00000000
ds 0x----0010
es 0x----0010
fs 0x----0000
gs 0x----0000
trap 0x0000000e Page Fault
err 0x00000002
eip 0xc0109eb1
cs 0x----0008
flag 0x00000086 PF,SF,IPL=0
kernel panic at kern/trap/trap.c:213:
  handle pgfault failed in kernel mode. ret=-3

Welcome to the kernel debug monitor!!
```

然后发现是 load\_icode 被改过了，要完成 Lab8 编码才能正常 work…有点坑

## Ex1

更新了 proc.c 对 process 数据结构的初始化，do\_fork，trap 中的一些代码

完成了 sfs\_inode 中 sfs\_io\_nolock 函数的编码，分为了几种情形：

起始数据块是否完全被覆盖；中间数据块有无完全被覆盖的数据块；结束数据块是否完全被覆盖。

可以通过 make run-sfs\_filetest1

请在实验报告中给出设计实现“UNIX 的 PIPE 机制”的概要设方案,鼓励给出详细设计方案。

#保留一部分磁盘空间作为缓冲区

#缓冲区拥有头，尾，等待写入队列，等待读取队列

#创建管道时完成初始化

#进程使用管道，使用计数器的方式描述

#当使用计数为 0 时回收管道

#读写数据均使用逐字节的方式

#读取数据利用缓冲区，缓冲区非空时，唤醒写入等待队列的进程，缓冲区为空时，该进程阻塞，加入读取等待队列，等待唤醒

#写入数据同上，缓冲区满时，进程阻塞，加入写入等待队列，缓冲区不满时，写入一个字节，唤醒读取等待队列

## Ex2

编码过程中参考了之前 lab7 的 load\_icode 以及 labanswer...

不过在 make qemu 时跑通所有测试后 initproc 退出了..无法测试支持文件系统的命令行, 不知道原因..

请在实验报告中给出设计实现基于”UNIX 的硬链接和软链接机制“的概要设计方案, 鼓励给出详细设计方案

硬链接: 创建时目录项名字为传入参数, inode 为目标文件的 inode, 同时 inode 的引用计数加一; 删除时目标的引用计数减一, 若为零则回收。

软链接: 创建时 inode 类型为符号链接, 数据为目标路径; 删除时无需操作。