

## 修改内容

- 合并与框架bug修复：sys\_write等对fd的上界进行检查；sys\_write将fd==0改成fd<=2；loader.c中将names从20改成30
- 定义调用宏SYS\_mail\_read, SYS\_mail\_write, 实现由pid获取proc\*的函数（通过遍历）
- proc中添加mail结构，采用类似pipe的ring buffer管理机制，并添加相关初始化
- 按要求实现sys\_mail\_write和sys\_mail\_read

## 思考题

1、ps -aux | grep python 可以获取目前正在运行的python应用

### 2.1.1、多核的问题

写写冲突：两个进程向一个进程A发邮件，由于没有保护它们向同一个位置写入了邮件，因此进程A只收到1封邮件

读写冲突：在进程A向进程B写邮件时进程B正在读，会出现读到了写了一半的内容或者还没写的内容

读写冲突：进程A读邮件时进行B向进程A写邮件，此时如果A的邮箱满了会导致进程B写失败（但实际上通过读邮件已经使邮箱不满了）

### 2.1.2、单核的问题

在我们的操作系统内不会，因为内核代码屏蔽了中断，因此sys\_mail\_write和sys\_mail\_read均为原子操作。用户使用时可能会出现“测试是否能写”——“返回能”——“尝试写”——“写失败，邮箱满”的情况，但这不属于操作系统的问题，用户应当能处理这种情况。

2.2、使用m+n个互斥锁，每个进程的锁负责对应进程的邮箱，读邮件时申请自己的互斥锁，写邮件时申请对方的互斥锁，只有拿到互斥锁才能进行读写操作

2.3、如果每次读取都可能读取整个邮箱的所有内容，而不一定只读取一个报文的话，则使用ring buffer这种结构获取整个邮箱的内容会有些麻烦，将其修改成queue的结构可能会更好