

思考题

- 5.1
 - 在Linux中有额外的机制可以为内核和内核模块将信息发送给进程-- /proc 文件系统。最初设计的目的是允许更方便的对进程信息进行访问（因此得名），现在它被每一个有有趣的东西报告的内核使用，例如/proc/modules 有模块的列表/proc/meminfo 有内存使用的统计表。
 - Windows通过Win32 API的调用实现类似的功能。
 - 这样的设计使用户层面将内核视为文件，简化了交互过程。
- 5.2
 - 会因为cache的写回机制产生问题

具体来说，cache是在被置换的时候写回的。而console这种外设具有实时交互性。如果一个向console的输出想要通过cache实现，那么只有在这块cache被替换的时候，才能真正在console上看到对应的输出，这十分荒谬。另一个例子是，例如sys_dev_read对于0x30的“检验”操作，在上述描述下将永远返回false，更加荒谬
- 5.3
 - $\text{Byte} * \text{BY2BLK} * 1024 = 4\text{MB}$
- 5.4
 - $\text{BY2BLK} / \text{sizeof}(\text{struct File}) = 16$
 - $16 * 1024 = 16384$
- 5.5
 - 表面上看，是DISKMAX=3G空间
 - 实际上，是ULIM-DISKMAP=1920M（因为fs进程是一个用户进程）
- 5.6
 - 首先，先看一下Open时的过程：

fd是一种资源，独占1页的起始。open的时候，将open的需求和申请到的空fd资源通过任务号（value）和IPC的传递页面一起发送给文件进程。文件进程那边通过一系列行为，对这一页进行了很多改动。不仅是fd所代表的那12字节，而包括后面的许多字节。共享解除后，用户这边的这一页上已经写好了全部必要信息，而fd不过是一个首地址，强转为filefd后还是那个首地址，区别仅在于访问方式更加多（可以访问属于filefd的，后面那些结构体成员）
 - 既然这一页已经全部写好，那么属于filefd的那一部分也被写好了，也可以访问。
 - 事实上，fd和filefd类似一个extend的关系，filefd这个结构体一开始是一个fd，然后是一些别的内容。在内存排布上就是一个extend的关系。
- 5.7
 - File用来描述文件，其中
 - name是文件名
 - size是内容大小
 - type是类型（dir, normal, 5-2-Extra出现的Ink）
 - direct是一开始那10个块
 - indirect是间接部分那些block的地址
 - dir是这个文件的目录的地址，也就是".."
 - pad是填充物，目的是凑成一个FILE结构体恰好256Byte
 - fd用来以较小的代价描述一个文件，独占1页,但仅需12字节

- devid是外设id，也就是外设类型
- offset是读或写的当前位置（偏移量），类似于“流”的当前位置
- omode是打开方式，比如只读，只写，均可blabla
- Filefd是真正的描述一个文件，代价较大（16+256==272字节），包含了整个struct File，就出在fd独占的那一页
 - fd就是之前提到过的fd。用fd描述文件就像是拉着领头羊（fd）控制羊群（后面地址保存的File）一样
 - fileid就是file的id，顾名思义
 - file就是真正的文件，相当于藏在fd后面的真正的资源。然而大多数时候只需要fd的信息就足够了
- Open是抽象了打开某个文件这一行为，同样的，代价较小（想要精确的话，直接用struct File即可，但是代价较大）
 - file指向具体的File
 - fileid同上
 - omode同上
 - off同上

可以看到，Open结构体的内容之前都出现过，之所以抽象出来，可能是为了方便操作吧。不过仅看lab5这些功能，似乎有些过度封装？

• 5.8

- 这个serve是一个用户进程，相当于提供服务的“后台进程”
- serve死循环了，说明没有用户目前需要服务，那么当时间片到的时候，就会被替换成用户进程继续跑
- 类似java的思路，后台进程死亡当且仅当所有非后台进程死亡，也许lab5中也有类似的机制但我并没有发现（事实上OS课的小操作系统并没有关机的能力？）

难点图示

- 总体来说，lab5有关MIPS的一部分还不太困难，主要由三大方面：
 - 块缓存与外设
 - 文件目录与具体文件的组织形式
 - fsipc的服务

其中，第一部分类似lab2，第二部分类似lab2的虚存，第三部分类似syscall

- 难点之一在于有关Linux的一部分（创建fs.img）
- 难点之二在于lab5建立在之前的基础上，容易出现陈年老bug

体会与感想

- 不知为何，我感觉lab5体验比lab4总体来说要好不少，可能是因为lab5很多内容我没有细想
- 在lab5刚开放的时候，我就迅速抄完了代码，按照自己的理解改了改，通过了评测
- 之后的主要任务便是理解上一部分所说的三大块内容，逐个击破之后，lab5的理解难点也基本扫清了