

操作系统lab5实验报告

17373452 单彦博

一、思考题

5.1 查阅资料，了解Linux/Unix 的/proc 文件系统是什么？有什么作用？Windows 操作系统又是如何实现这些功能的？proc 文件系统这样的设计有什么好处和可以改进的地方？

/proc文件系统是LINUX/UNIX系统上的一种内核和内核模块用来向进程 (process) 发送信息的机制。其作用是提供了一种和内核内部数据结构进行交互、获取有关进程的有用信息、在运行中改变设置(通过改变内核参数)的方式。Windows系统通过Win32 API的函数调用来完成与内核模式的交互。这样的设计将对内核信息的访问交互抽象成了对文件的访问修改，简化了交互过程。

5.2 如果我们通过kseg0 读写设备，我们对于设备的写入会缓存到Cache中。通过kseg0 访问设备是一种错误的行为，在实际编写代码的时候这么做会引发不可预知的问题。请你思考：这么做这会引起什么问题？对于不同种类的设备（如我们提到的串口设备和IDE 磁盘）的操作会有差异吗？可以从缓存的性质和缓存刷新的策略来考虑。

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。

5.3 一个Block 最多存储1024 个指向其他磁盘块的指针，试计算，我们的文件系统支持的单个文件的最大大小为多大？

$4KB * 1024 = 4MB$

5.4 查找代码中的相关定义，试回答一个磁盘块中最多能存储多少个文件控制块？一个目录最多能有多少个子文件？

一个磁盘块有FILE2BLK=16个文件控制块。一个目录最多指向1024个磁盘块，每个磁盘块下面最多有16个文件控制块，因此一个目录最多有 $1024 * 16 = 16384$ 个子文件。

5.5 请思考，在满足磁盘块缓存的设计的前提下，我们实验使用的内核支持的最大磁盘大小是多少？

最大磁盘块大小为DISKMAX = 3MB

5.6 阅读user/file.c 中的众多代码，发现很多函数中都会将一个struct Fd * 型的指针转换为struct Filefd * 型的指针，请解释为什么这样的转换可行。

在user/fd.h中，我们可以找到Fd和Filefd的定义，可以看到Filefd结构体中第一个元素就是一个Fd结构体，所以做这样的强制转换，就相当于将一个Filefd里面的Fd的内容赋值好，以便于接下来使用。

5.7 请解释File, Fd, Filefd, Open 结构体及其各个域的作用。比如各个结构体在哪些过程中被使用，是否对应磁盘上的物理实体还是单纯的内存数据等。说明形式自定，要求简洁明了，可大致勾勒出文件系统数据结构与物理实体的对应关系与设计框架。

1. File定义在include/fs.h，用来描述文件：

- name是文件名
- size是内容大小
- type是类型
- direct[NDIRECT]是最开始的10个直接指针
- indirect是间接指针，指向一个磁盘块
- dir是这个文件的目录的地址
- pad是填充物，目的是凑成一个FILE结构体恰好256Byte

2. fd定义在user/fd.h，用来描述一个文件的基本信息，独占1页，但仅需12字节：

- dev_id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，读写等

3. Filefd定义在user/fd.h，是描述一个文件的详细信息，代价较大（16+256==272字节），包含了整个structFile，位置就在fd独占的那一页

- fd就是之前提到过的fd
- 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

4. Open定义在fs/serv.c，是抽象了打开某个文件这一行为

- file指向具体的File
- fileid就是file的id
- omode是打开方式，比如只读，只写，读写等
- offset是读或写的当前位置（偏移量）

5.8 阅读serve函数的代码，我们注意到函数中包含了一个死循环for (;;) {...}，为什么这段代码不会导致整个内核进入panic 状态？

这个serve是一个用户进程，相当于提供服务的“后台进程”，它会在操作系统运行的时候一直提供文件的服务。

二、实验难点

个人认为lab5没有什么特别难以理解的内容，但是lab5唯一难点就是读懂代码，lab5的新增代码量加起来应该有lab1到lab4的量那么多，特别是文件操作相关代码，如file.c，fd.c，以及fs文件夹下的所有代码，想要读懂和理解他们之间的关系是需要花费很多时间的。

三、实验感想

本次实验的感觉很良好，在课下我花了十几个小时的时间。在课下花了很大功夫的基础上，我对小操作系统的文件管理系统有了很好的理解，并对其中所运用的方法表示惊讶，例如表驱动管理等等。以我现在的的能力，只能读懂这些代码，但是要我真真正去写这样的代码的话，还差的很远，我的能力还是有限，学习之路还很漫长。

在课上测试中，我完成了两次的基础题，并完成了第一次的附加题，通过学习文件系统，我对它的理解不再停留在理论课讲的表面内容，而是真正知道怎么去实现了。而且通过做实验，对理论课所讲知识有着很好的复习作用。

操作系统实验也马上接近尾声了，在之后的lab6实验中，我还会继续努力去做好每一个细节。