# 同济大学计算机系操作系统课程作业 进程管理 一 2023-11-13

学号 2151769 姓名 吕博文

一、（1）注释PPT10~11，写出read系统调用的执行细节（2）画2张图，补全随后write系统调用的执行细节。不必面面俱到，不清楚的地方红笔标出来，本周四前完成。

文本

描述已自动生成

日程表

描述已自动生成

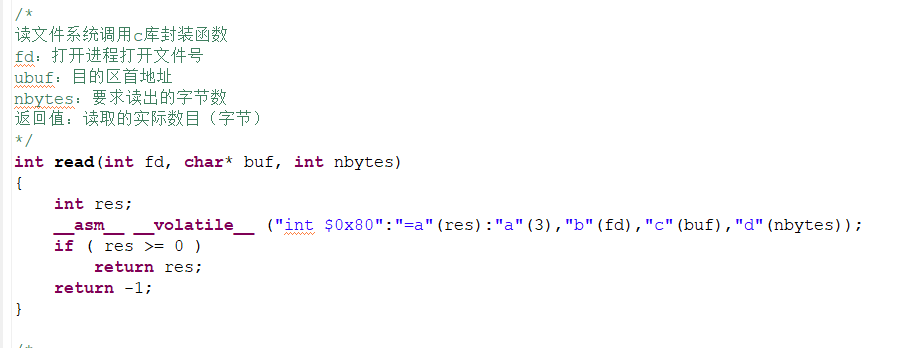


日程表

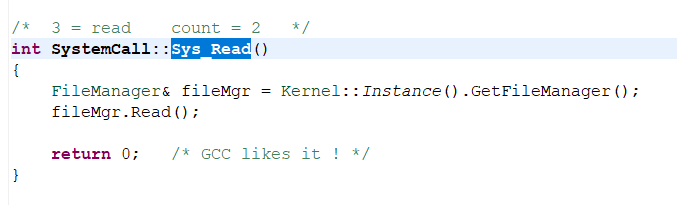
描述已自动生成

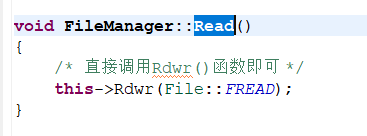
1、

1. read函数的UNIX V6++实现如下：



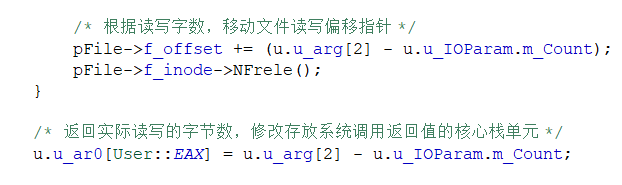
系统调用Sys\_Read函数如下:











1. 系统调用Read的具体实现细节：

在具体的read（）函数实现中，首先，根据系统调用参数fd获取打开文件控制块结构，并判断文件打开是否成功，读写模式是否正确，之后，首先记录目标缓冲区首地址，要求读入的字节数，以及读入内容送达地址，之后根据是否为管道读写分类实现，如果是管道读写直接调用函数读入即可，如果是普通文件读写或是特殊文件，则对普通文件或特殊文件实施互斥访问，通过调用文件对应的Inode的NFlock和NFrele方法实现。设置文件起始读位置为文件控制块的偏移量（f\_offset）。如果是读操作，调用Inode的ReadI方法；如果是写操作，调用Inode的WriteI方法。

pFile->f\_offset += (u.u\_arg[2] - u.u\_IOParam.m\_Count);: 根据读写字节数，移动文件读写偏移指针。

pFile->f\_inode->NFrele();: 释放对Inode的锁定，表示文件读写操作完成。

u.u\_ar0[User::EAX] = u.u\_arg[2] - u.u\_IOParam.m\_Count;: 返回实际读写的字节数，修改存放系统调用返回值的核心栈单元（EAX寄存器）。

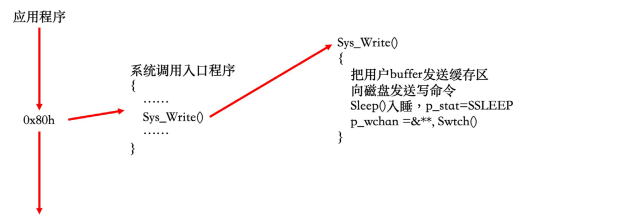
订正：

参与read系统调用执行的有两个硬件,CPU和磁盘，read系统调用分四个阶段，分别对应时序图的1、2、3、4，

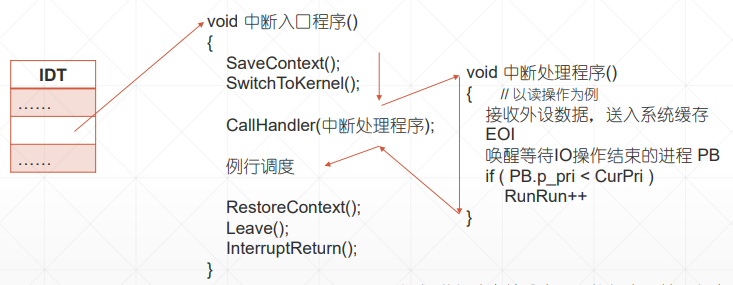
1. CPU侧，T0时刻，执行read系统调用的PA进程陷入内核，执行系统调用上半段，向磁盘发出IO命令。上图，对应红色字体的部分，完成后Sleep并设置入睡优先数放弃CPU
2. 磁盘硬件：（T0，T1）时段执行IO操作，读取文件数据。完成之后向CPU送磁盘中断请求
3. CPU侧：T1时刻，响应磁盘中断，先运行进程PB陷入内核执行磁盘中断处理程序，读取磁盘硬件送来的文件数据，存入核心态内存，唤醒睡眠进程PA；完成之后PB中断返回，例行调度，PB被剥夺，执行Switch放弃CPU，PA优先级最高被选中，成为新运行进程，Switch返回。
4. CPU侧：PA进程Sleep返回，执行read系统调用的下半段，将核心态缓存中存放的文件数据送入用户空间buffer数组。

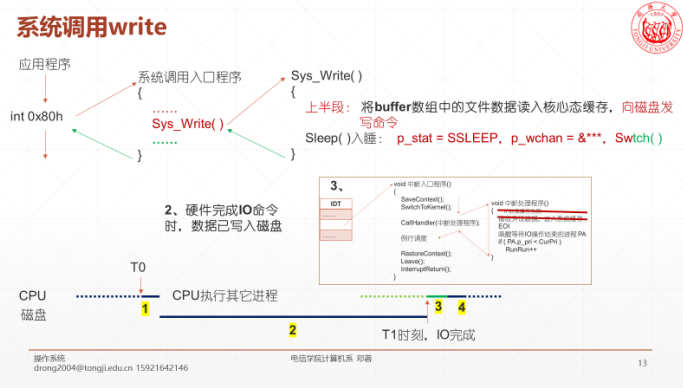
至此，read系统调用完成，PB返回用户态（可能会放弃CPU），执行write系统调用将buffer中的数据写入新文件。

2、应用程序使用int 0x80 进行系统调用，在系统调用入口调用Sys\_write()执行磁盘命令，随后该进程睡觉，切换进程



写任务完成之后，当前进程中断，进入对应的入口程序，在中断处理程序时唤醒进程并回送EOI信号，进行例行调度。





二、修改Kernel.cpp中的GetUser( )函数，用ESP寄存器计算得到现运行进程user结构的起始地址（调通系统和我说一声，加分）。评价系统性能。

图形用户界面, 文本, 应用程序

描述已自动生成

User结构与核心栈同时位于一个4K的字节块中，User结构的首地址位于该块的顶部，而此时EBP一定指向核心栈，所以EBP和User结构首地址一定共享高20位，而User结构首地址低12位一定是0，所以只需要将EBP的低12位清零即可。

