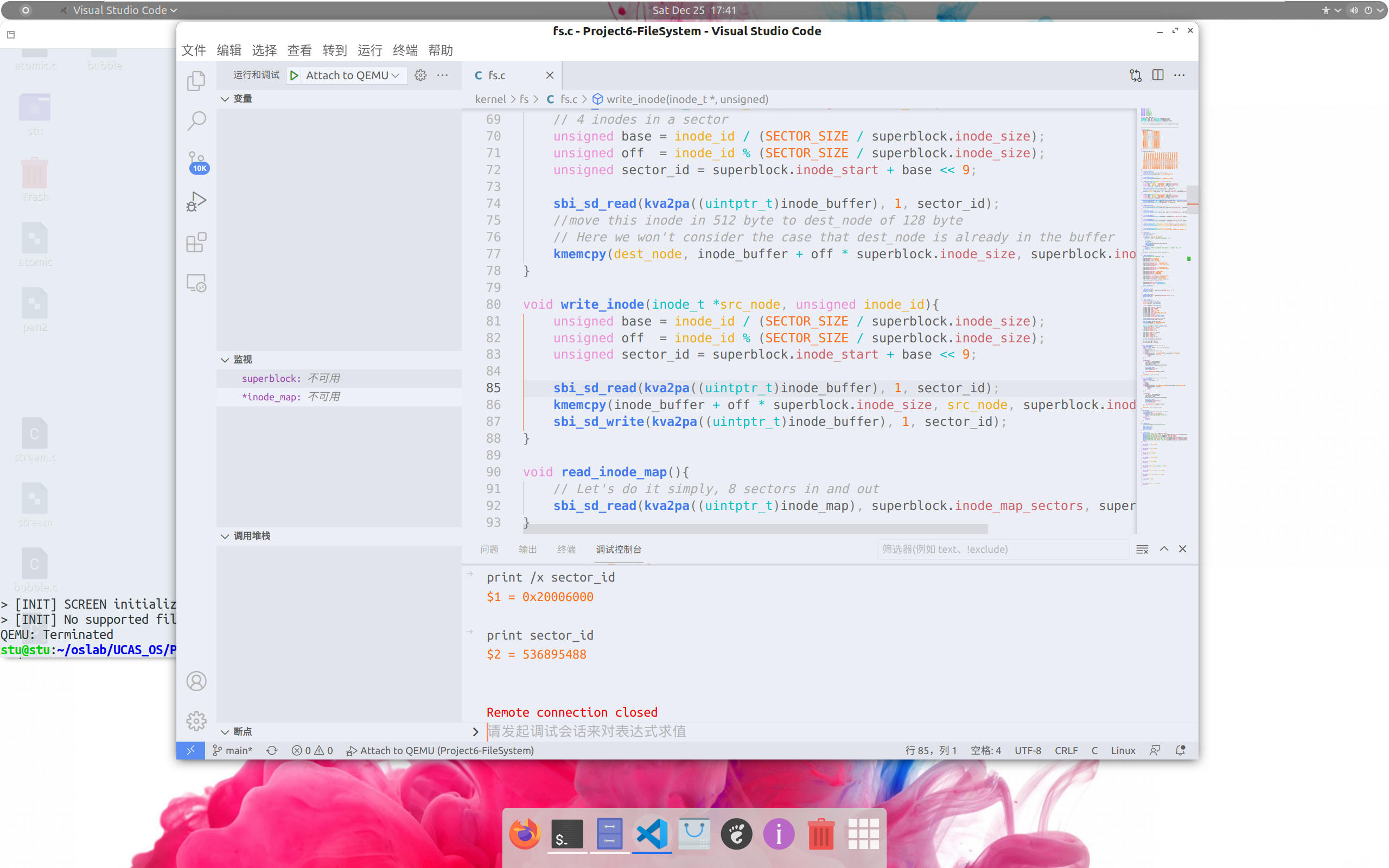
**操作系统研讨课 实验报告**

代瀚堃 2019K8009929051

**一、实验中遇到的问题**

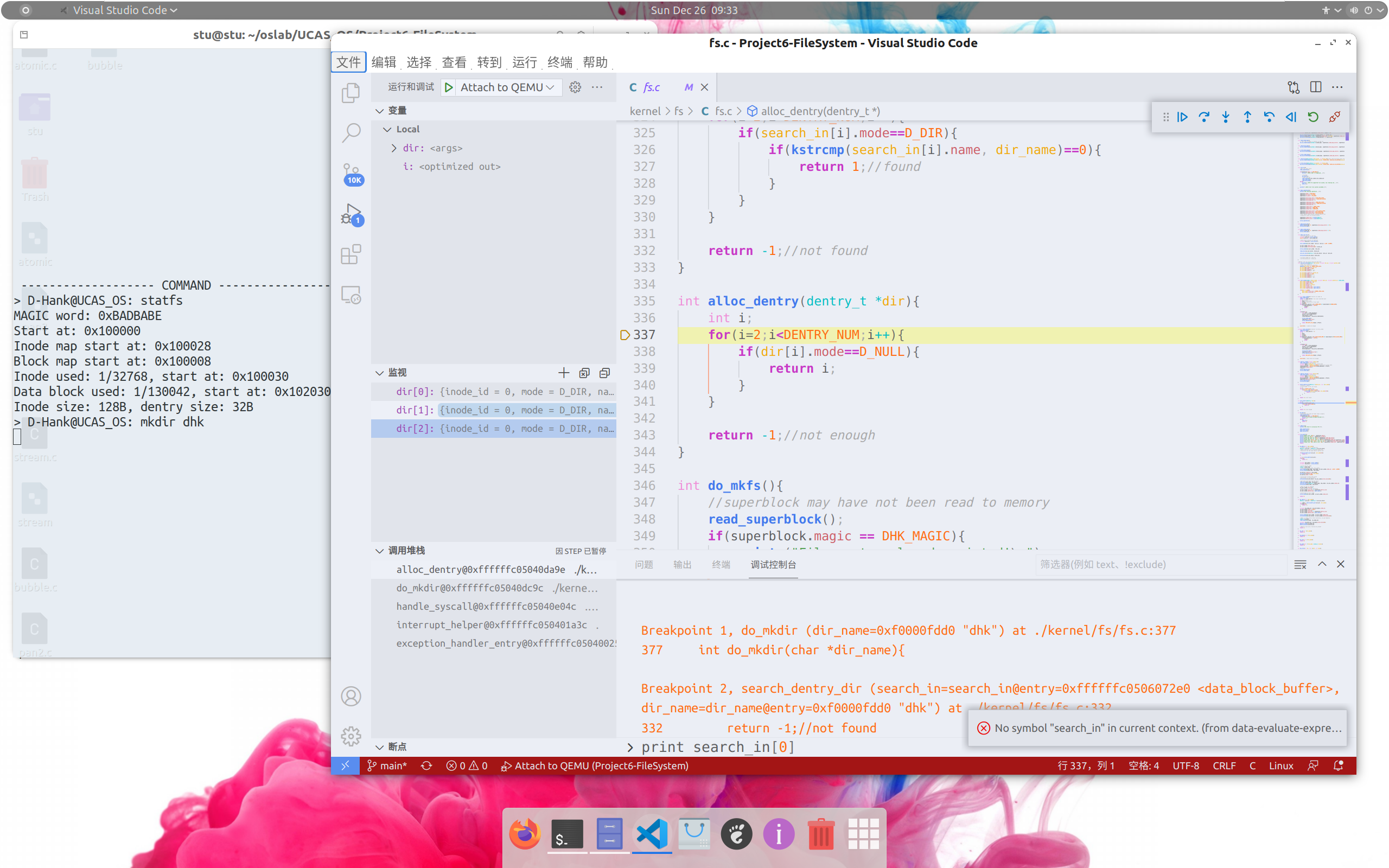
1. 读写SD卡出现block error

左移的优先级出错，导致块号变得很大：

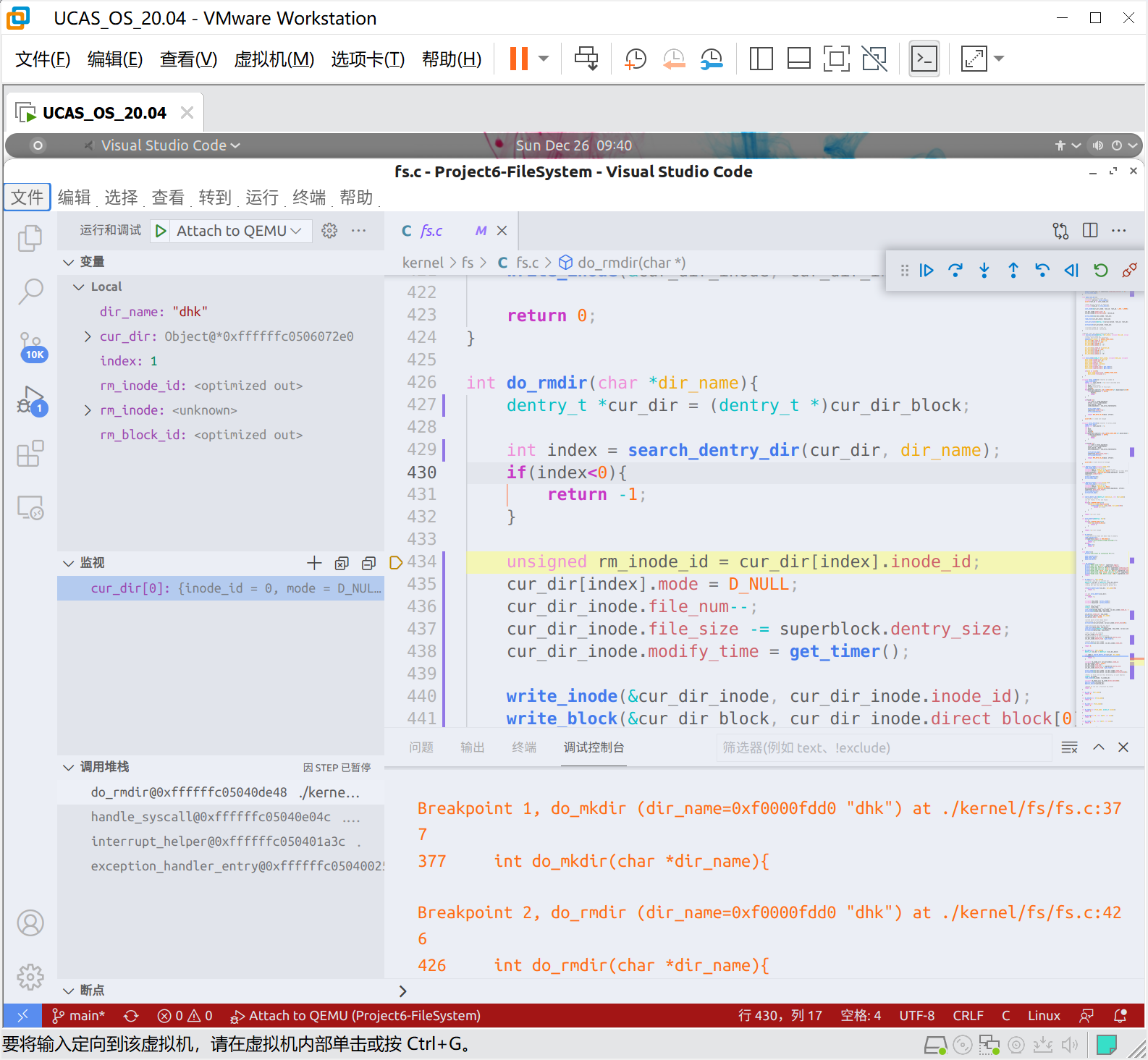


2. 表项或者inode中的某些项没有初始化

mkdir失败，原因是dentry的mode被初始化为I\_DIR：



在rmdir时，search\_dentry\_dir没有查找到，本来应该有的项变成了NULL：



可能是init\_dir\_block有问题，看了一下，确实在初始化的时候把类型弄错了：



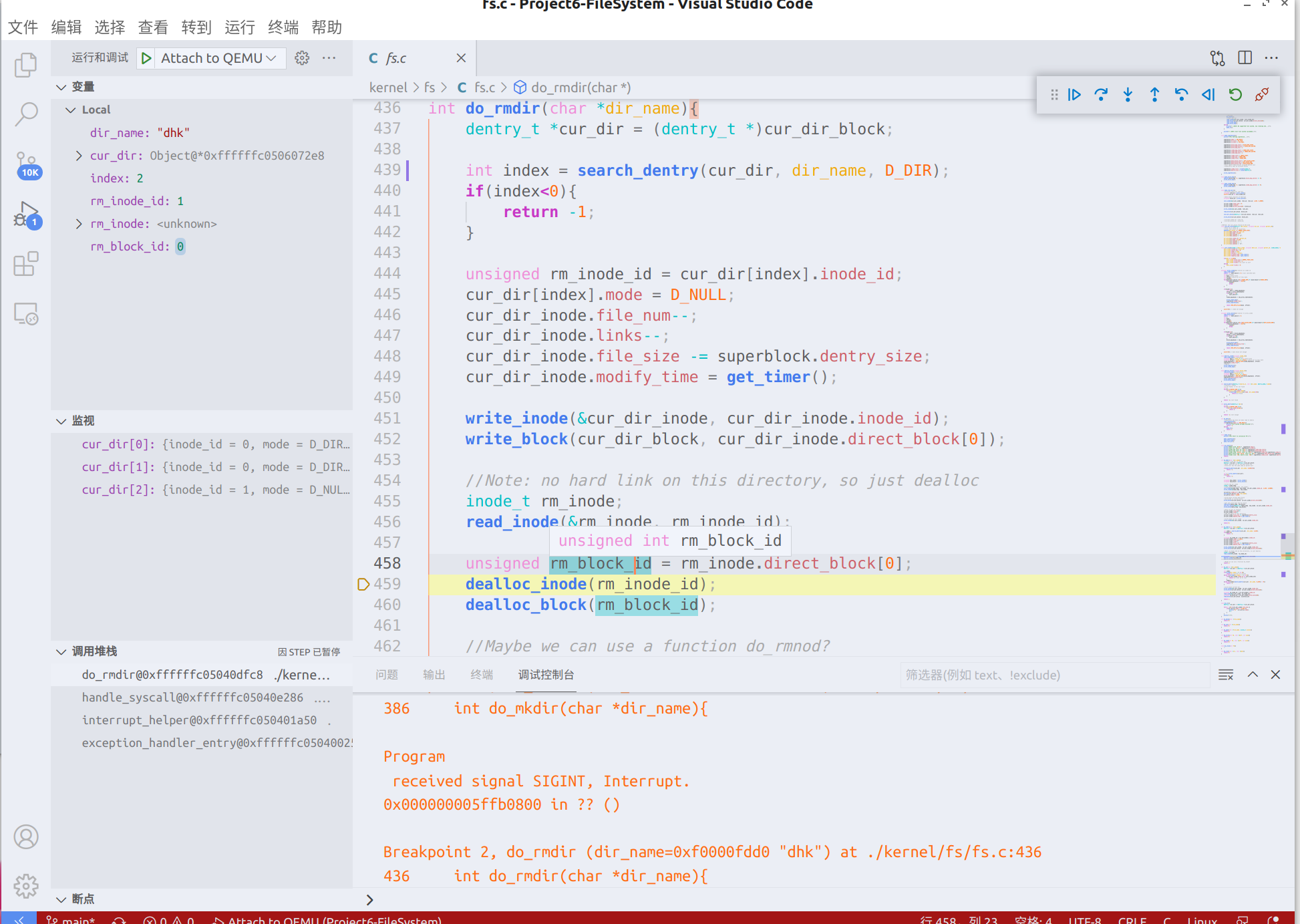
应该是D\_DIR

ls卡死，问题出在init\_inode这里，其步骤为：

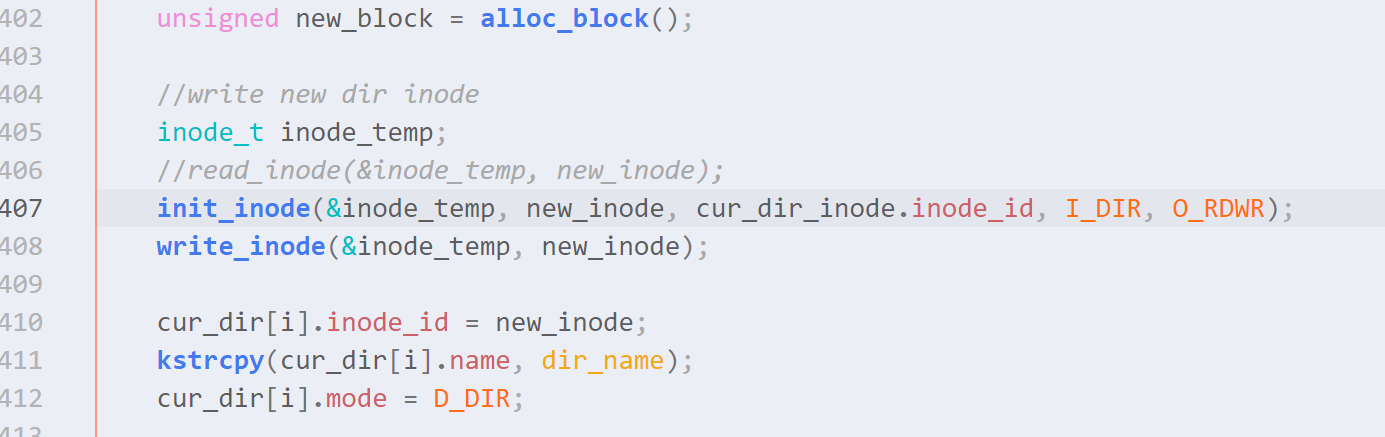
开机-mkdir dhk-rmdir dhk-mkdir dhk-关机-开机-ls

第一次mkdir：根目录第2项，inode=1, block=1，

rmdir的时候读回来的block就变成0了：



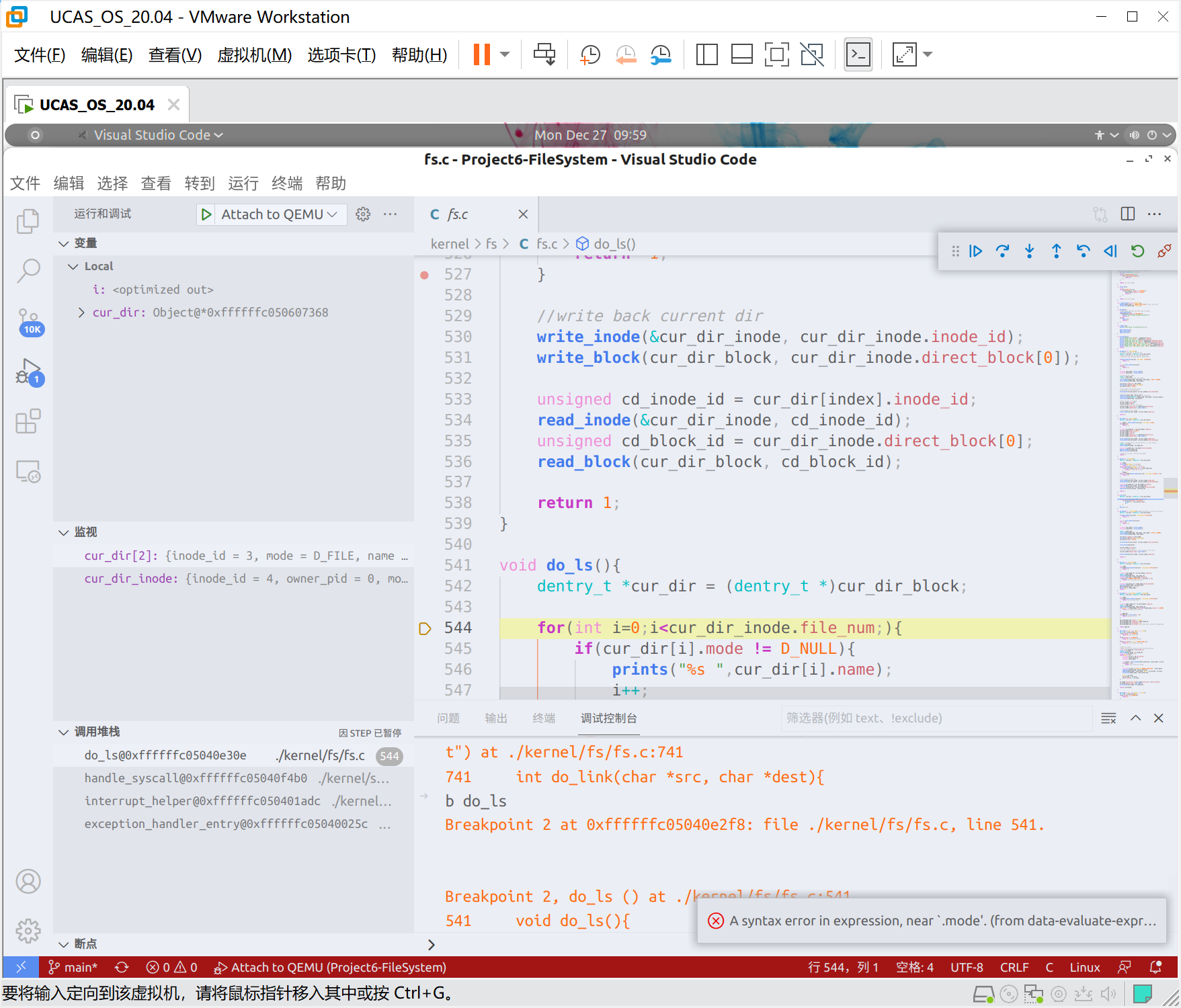
原因是在原来mkdir时，没有把block号存进去，写到磁盘上的还是inode\_temp内随机的数据：



init\_inode只完成了部分初始化的内容，其余内容（file\_size，file\_num，block）需要根据创建的是文件还是目录来确定

3. 块内的数据已经持久化到SD卡上，但没有同步到内存中的缓存

link之后ls查看test下并没有2.txt：

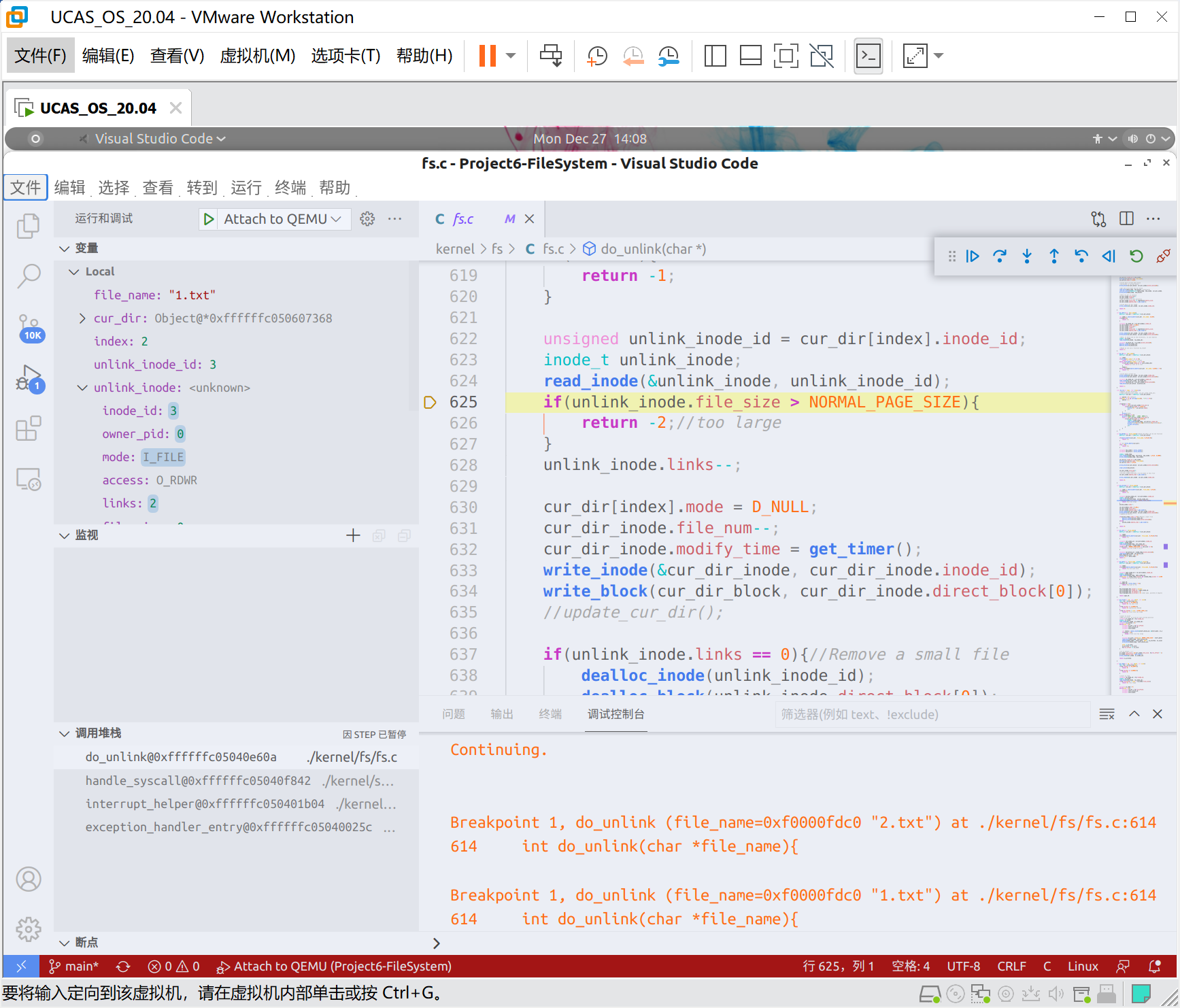


原因是link更新的是inode\_temp，并写到了block内，而缓存在mem中的当前目录并没有更新，所以每次改写当前目录后，都用update\_cur\_dir函数重新读一下当前目录的inode和数据块

rm之后没有删除干净，其过程为：

创建目录/dhk/ucas，创建文件/dhk/ucas/1.txt，返回至根目录，创建目录/test，切至test，创建链接ln /dhk/ucas/1.txt 2.txt，删除2.txt，返回至ucas目录下删除1.txt，然后statfs，占用块仍为5个

原因是没有把unlink\_node写回，后面读上来又是2了：



**二、还有待解决的问题**

1. 虽然我们已经实现了路径解析的函数get\_parent\_dir，但目前只有do\_link函数应用上，支持相对路径和绝对路径的查找，其他函数内没有调用这个函数，所以暂时只能支持相对路径

2. 由于缓冲区有限，cat命令只支持一个direct块内文件内容的显示

3. do\_unlink函数目前只支持删除一个direct块的文件，如果超过这一大小，会报错