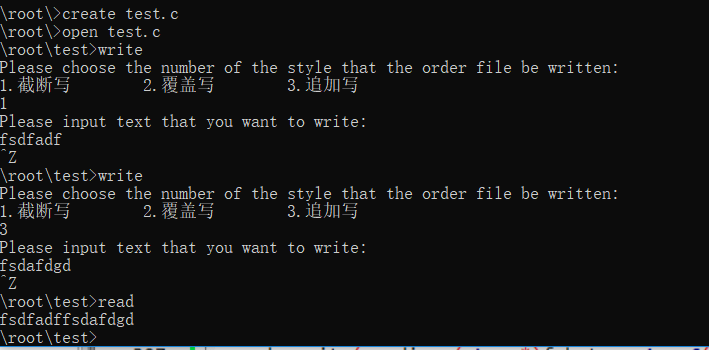
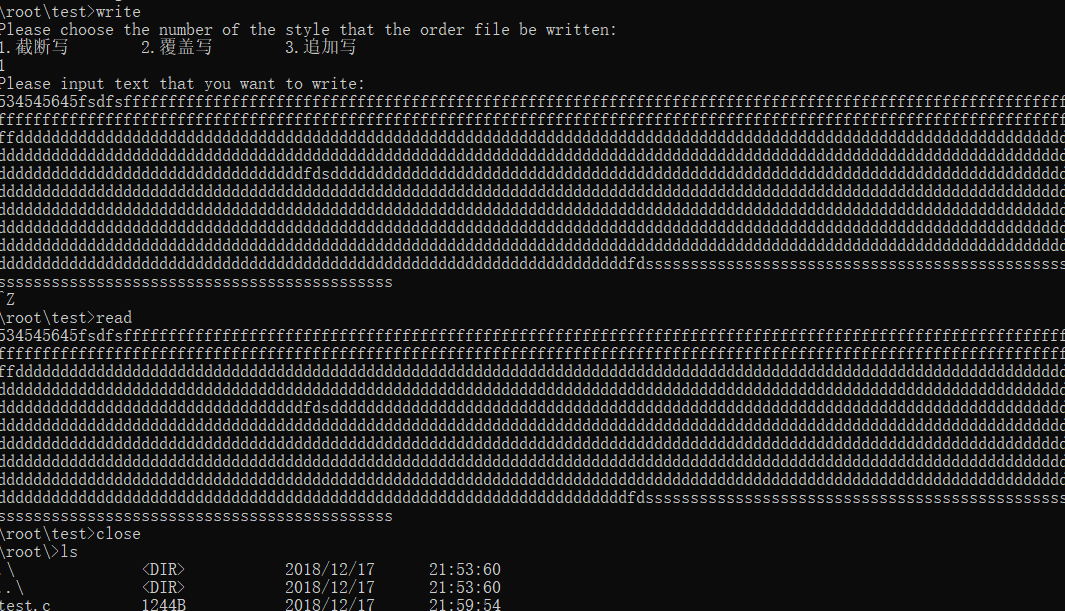


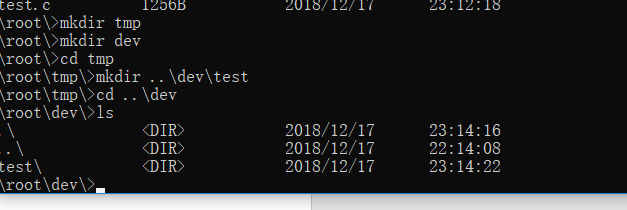
测试一下追加写



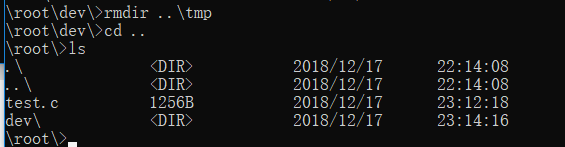
写入超过1024个字节



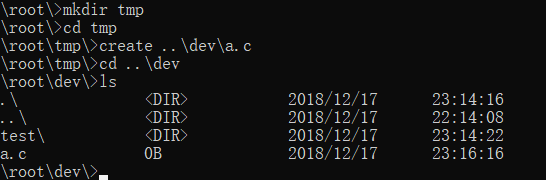
在指定目录创建目录



移除指定目录



在指定目录创建某个文件



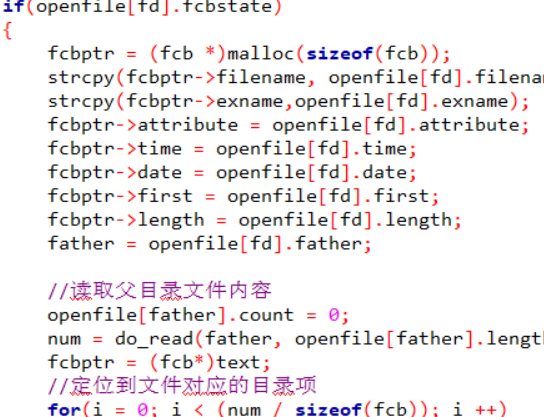
删除指定目录下的某个文件



**修改：**

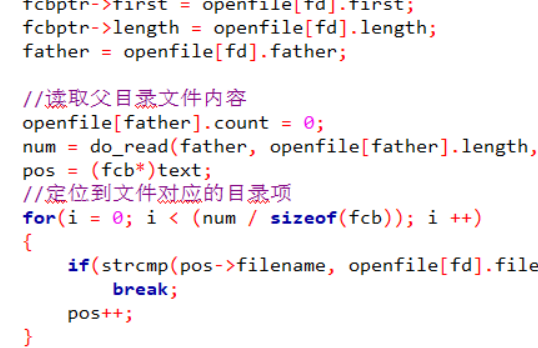
1.在创建的文件中写入东西，但是父目录的目录项中长度未更新，始终为0。

原因：my\_close()函数中，原代码为



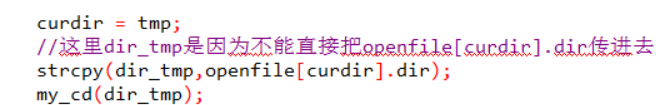
这里首先申请了一个fcb指针fcbptr，赋值后要把这个fcb写到父目录里，但是后面fcbptr又被覆盖了，所以写来写去，还是原来的东西写到父目录里去了。

修改：新定义一个fcb指针变量pos替换第二个fcbptr即可。



2.将my\_mkdir,my\_rmdir,my\_create,my\_rm中

开头：

结尾：

改成:

开头：strcpy(dir\_tmp,openfile[curdir].dir);

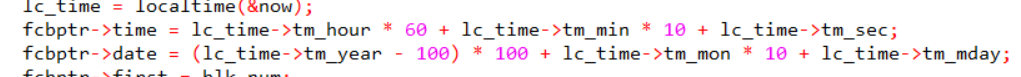
结尾:my\_cd(dir\_tmp);

因为tmp没有必要，仅保存路径即可。

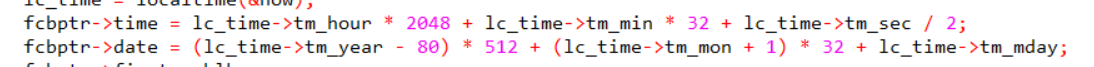
3.关于时间

（1）my\_mkdir()中对子目录fcb时间的初始化有问题

原代码：



改成：



（2）今天晚上讲的时候发现有些目录项时间是12月13号，是因为这个文件系统创建好保存到文件里的日期是12月13号。所以今天打开的是12月13创建的文件系统文件，里面的信息也是12月13号的。

4.关于my\_write的覆盖写

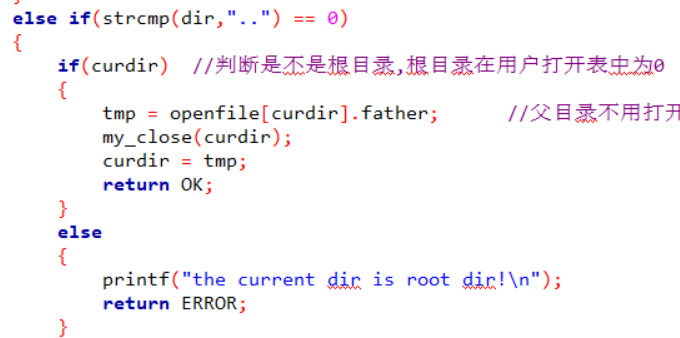
覆盖写不是没有实现，但不是在my\_write里实现的。每次在父目录中找到一个空闲的目录项来存放子目录fcb的时候，都是用的覆盖写的方式

比如：

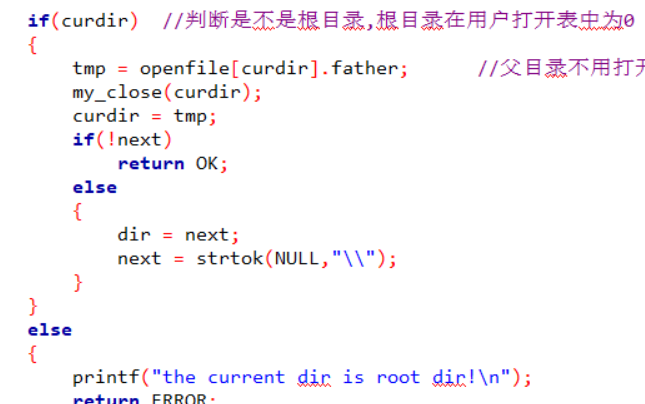
这里的2代表的就是覆盖写，从第i\*sizeof（fcb）开始对父目录中的内容进行覆盖。但这个参数2对于do\_write而言没有作用，只是象征着这是覆盖写而已。

5.my\_cd()中

原代码为：

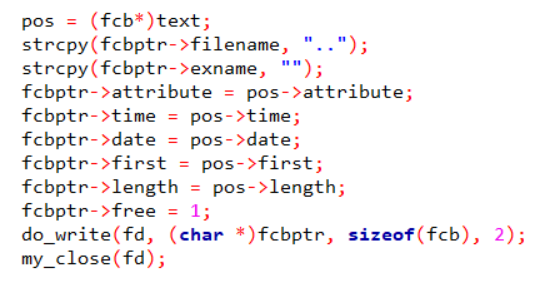


修改成：

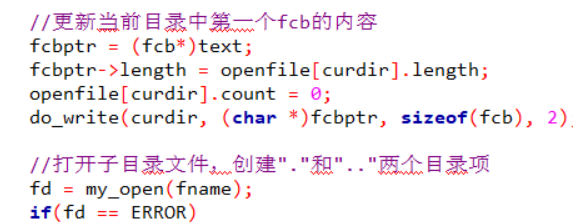


因为如果我输入的是cd ..\dev，那么按照源代码，切换到..(父目录)就直接函数结束return OK了，就不会切换到dev了。所以要判断一下next是不是NULL，如果不是NULL，还得进入后面的while循环继续切换。

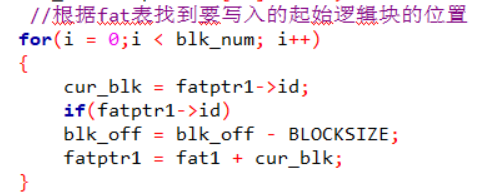
6.修改了my\_mkdir（）函数中写入子目录文件的“..”目录项的内容，这个目录项记录的是父目录的信息，所以要把父目录的信息写进去。



同时把父目录中第一个目录项的length更新放到创建子目录的”.”,”..”目录项之前。



7.do\_write函数中



删除if(fatptr1->id)

