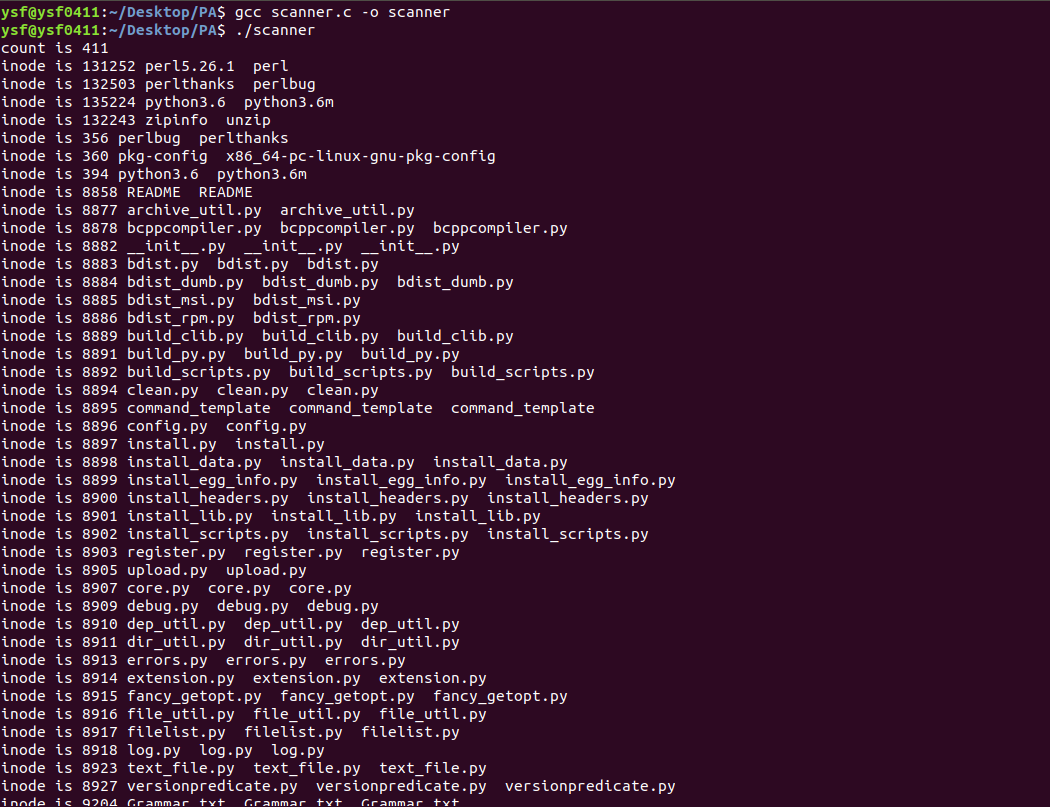
**编程作业实验报告**

191300073 杨斯凡 191300073@smail.nju.edu.cn

一、实验结果:



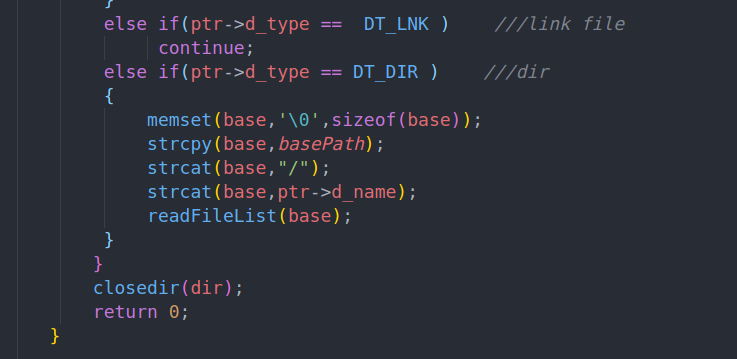
(scanner)

二、代码讲解

1. scanner



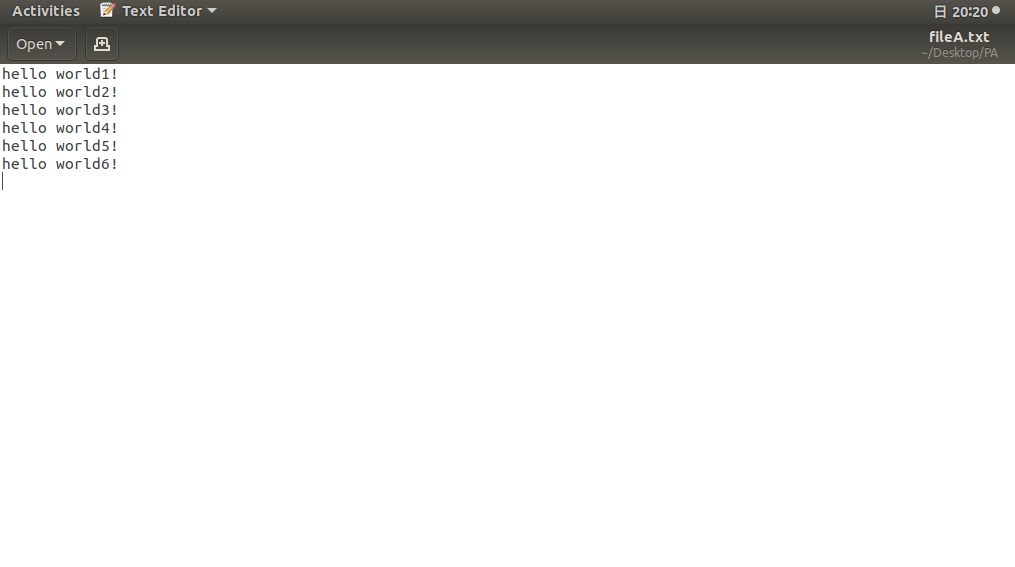
我使用了opendir函数来打开目录, readdir来进行读这个目录, 读到目录下的文件之后, 使用了stat这个API得到这个文件的inode信息, 判断连接数是否大于1, 如果大于1就将该文件名加入对应的字符数组中, 如果不是文件就递归向下读取即可.



最后输出所有的字符数组即可.

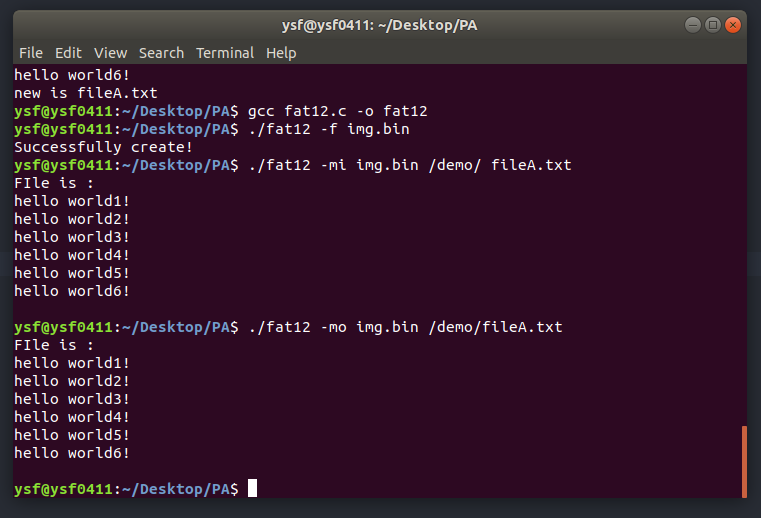
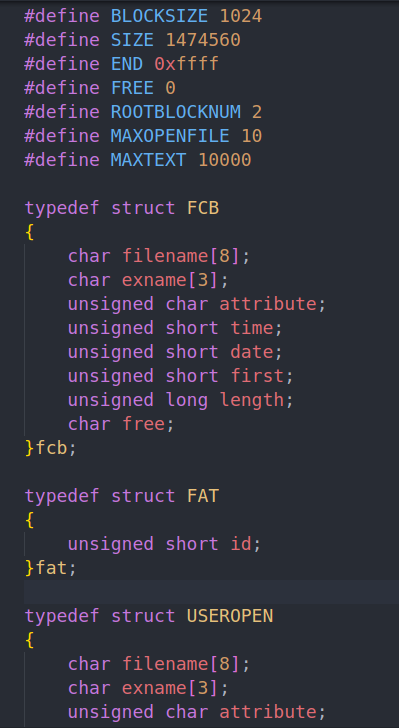
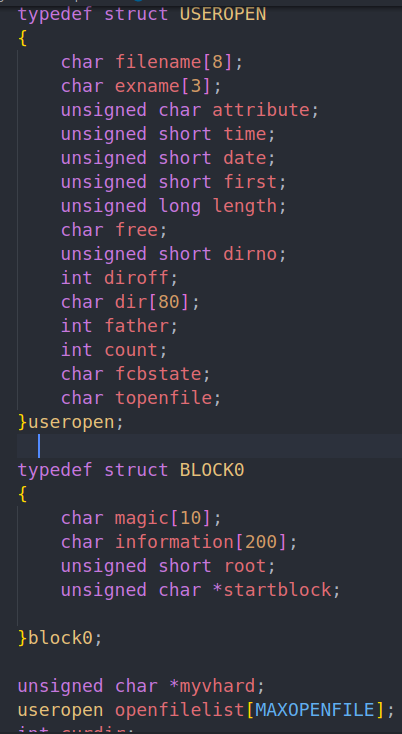
2. FAT12

为了让实验效果看起来更好, 我在每次文件复制结束之后把文件内容进行了打印.



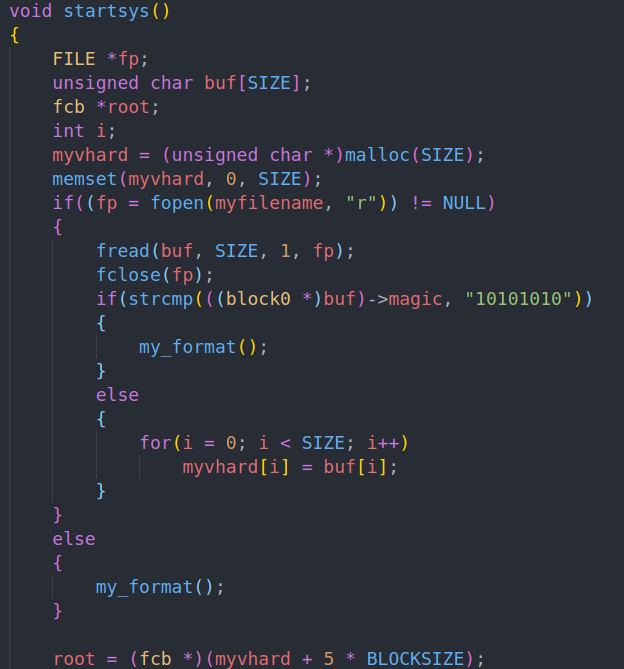
(这是原文件的内容, 最后一行有回车)

在这个编程作业中, 我首先阅读了FAT12的框架代码, 然后发现FAT12的簇定义需要12个字节, 实现太过于麻烦, 因此我使用了16个字节, 这样会导致数据区变小, 但是这样的速度却比12个字节块, 然后仿照FAT12的文件表, 超级块定义了一系列的结构体以供使用.



然后使用这些数据结构进行FAT12的模拟.

首先是创建:

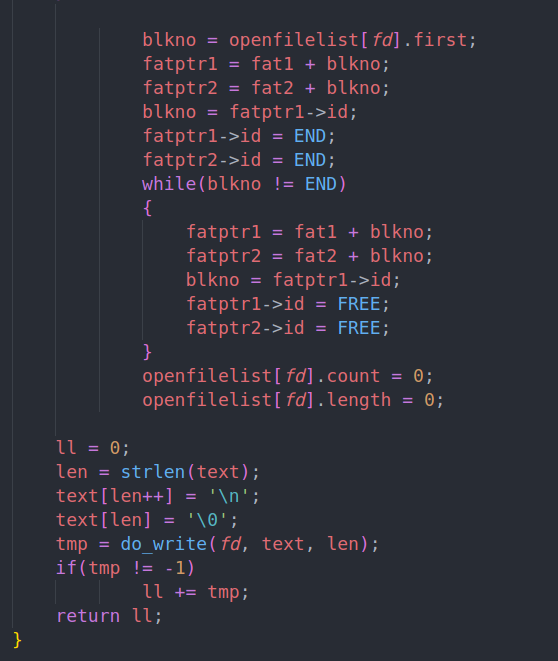
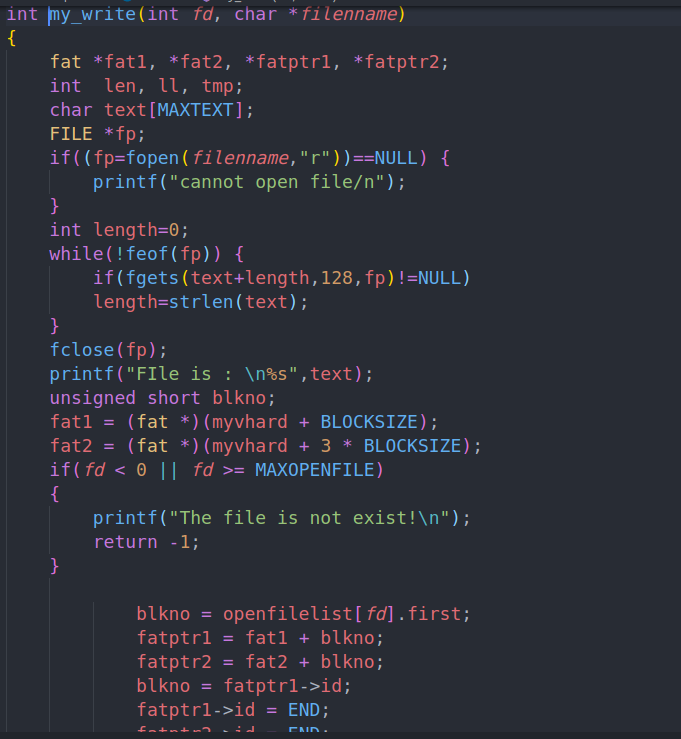


我会首先打开文件名的文件, 如果存在, 比较魔数是否相同，若不同，则需要格式化整个文件，如果相同，直接使用即可。

格式化文件，需要对文件的根目录和引导块进行设置，并且设置魔数以便于下次创建。

然后是复制进FAT12：

我会首先打开这个文件，然后将主机上的文件内容拷贝进text数组中，然后将这个数组内的内容写到我的FAT12文件系统中，这里如果没有文件会直接报错，这里需要主义的地方是需要对块的属性进行设置，以便于文件的打开。



然后是文件的复制出系统：

我首先打开了文件，然后还是同样的把文件内容读取到text中，然后把text的内容在主系统中写到新文件中即可。

