Lab6

实验报告

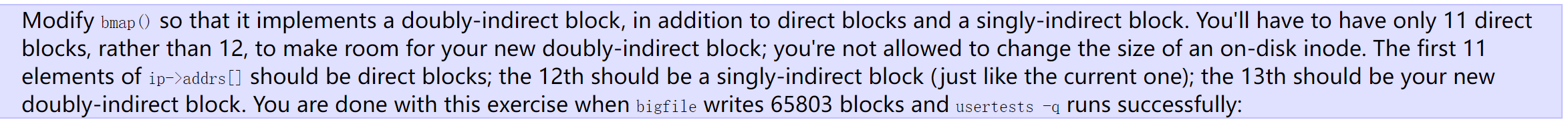
姓名 李晓畅

学号 20307130261

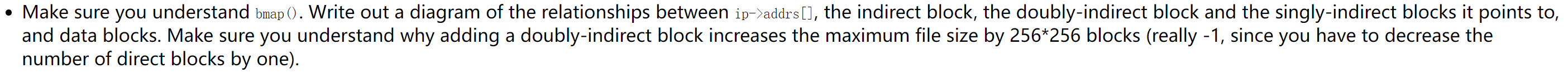
班级 计算机科学技术

# 实现思路

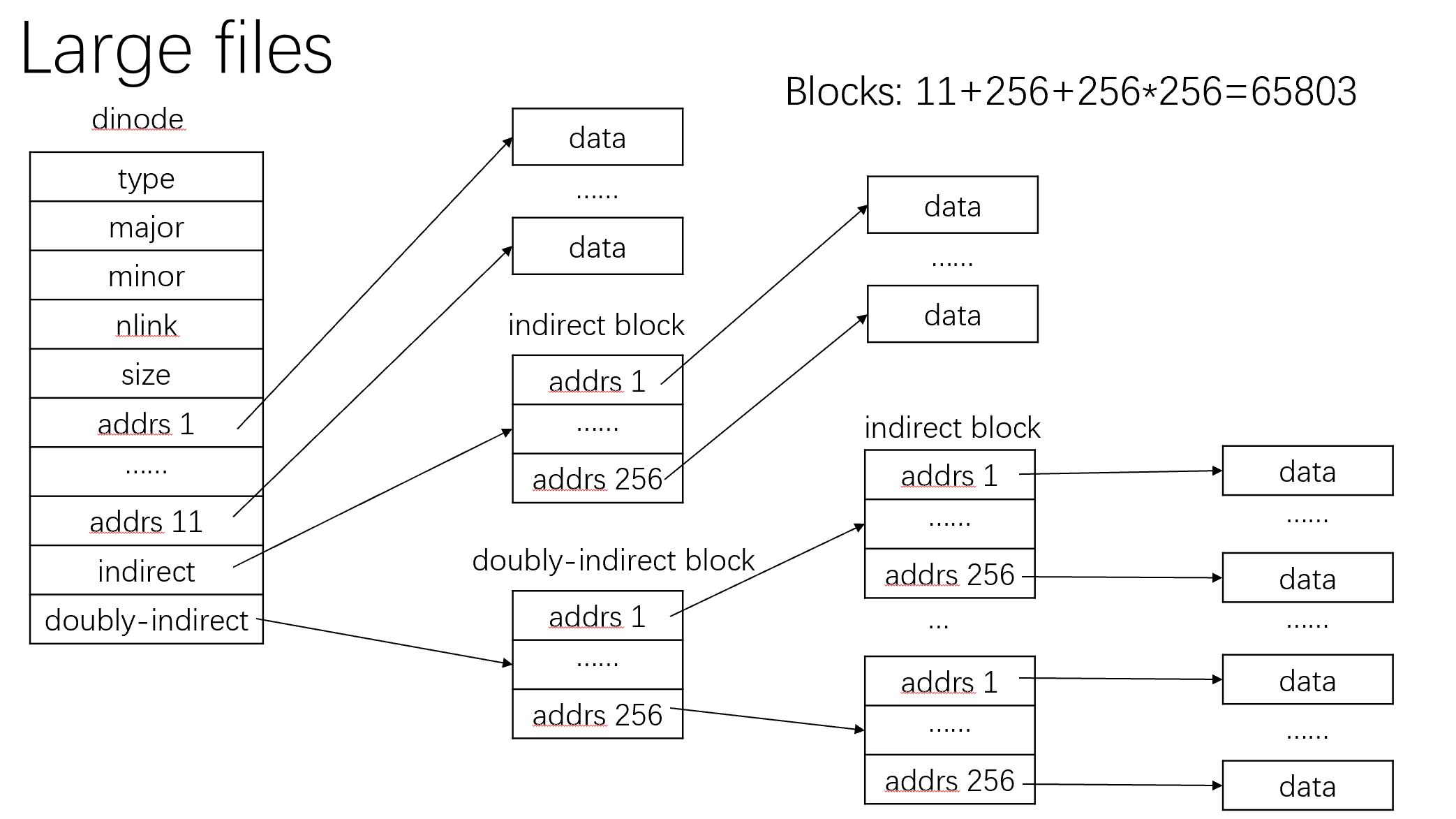
## **Large files**



按照提示进行：

* 

参考



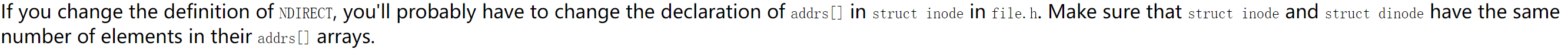
Ip->addrs[i]指的就是inode中第i个block编号。这些block里或许是数据，或许是索引，一级索引。

Indirect block中存放block号，这些block中存放数据。

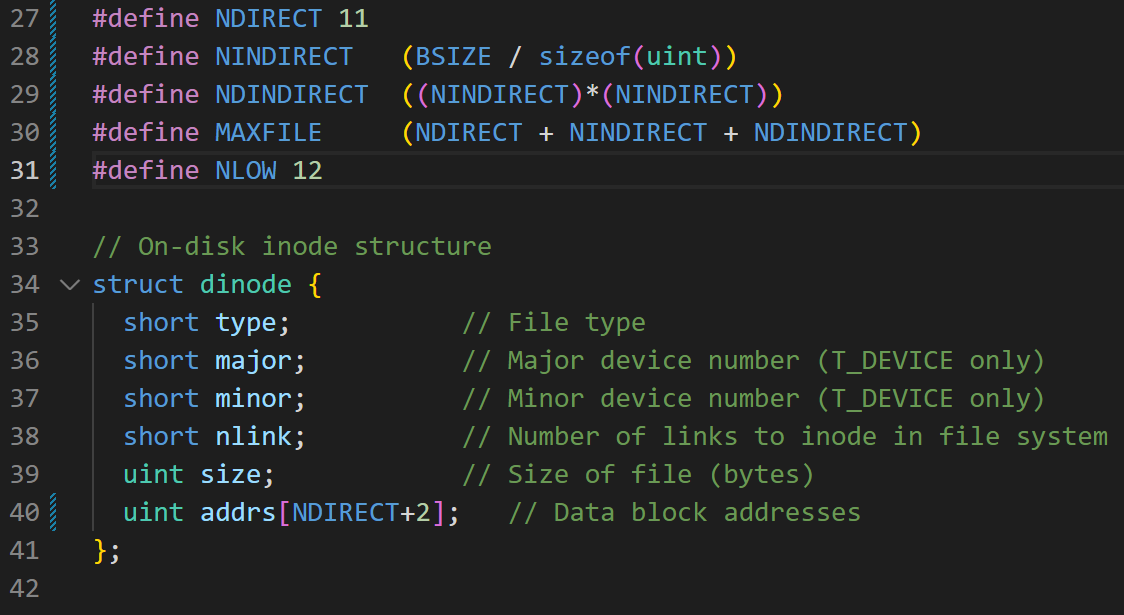
Doubly-indirect bloc中存放的是indirct block的块号。

每个索引包含256个entry，因为1024/4=256。

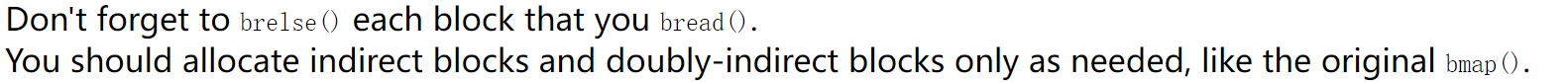
因而换掉一个直接块号，换成二级索引块号可以增加256\*256-1个块。

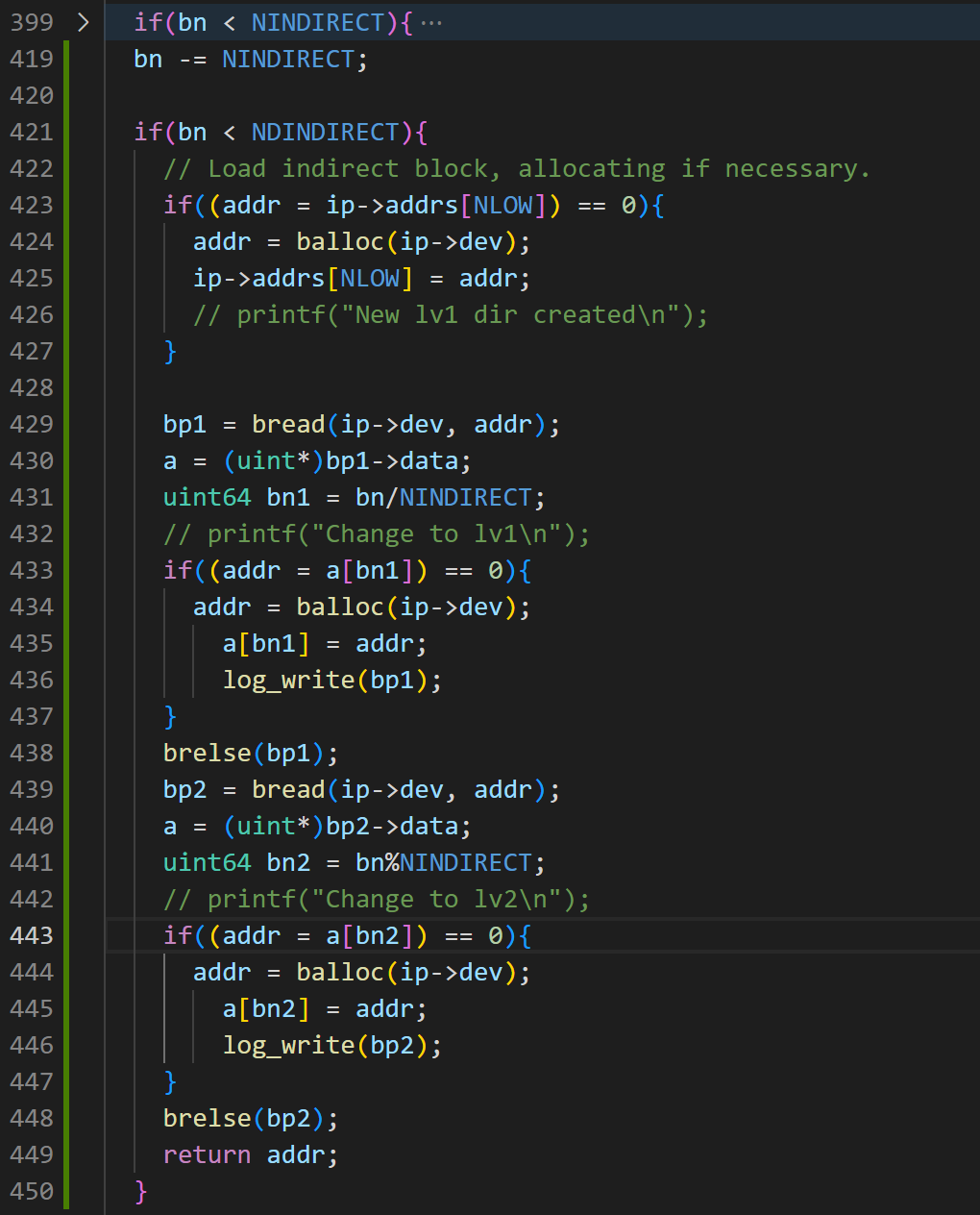
* 

对应地进行修改即可。





* 



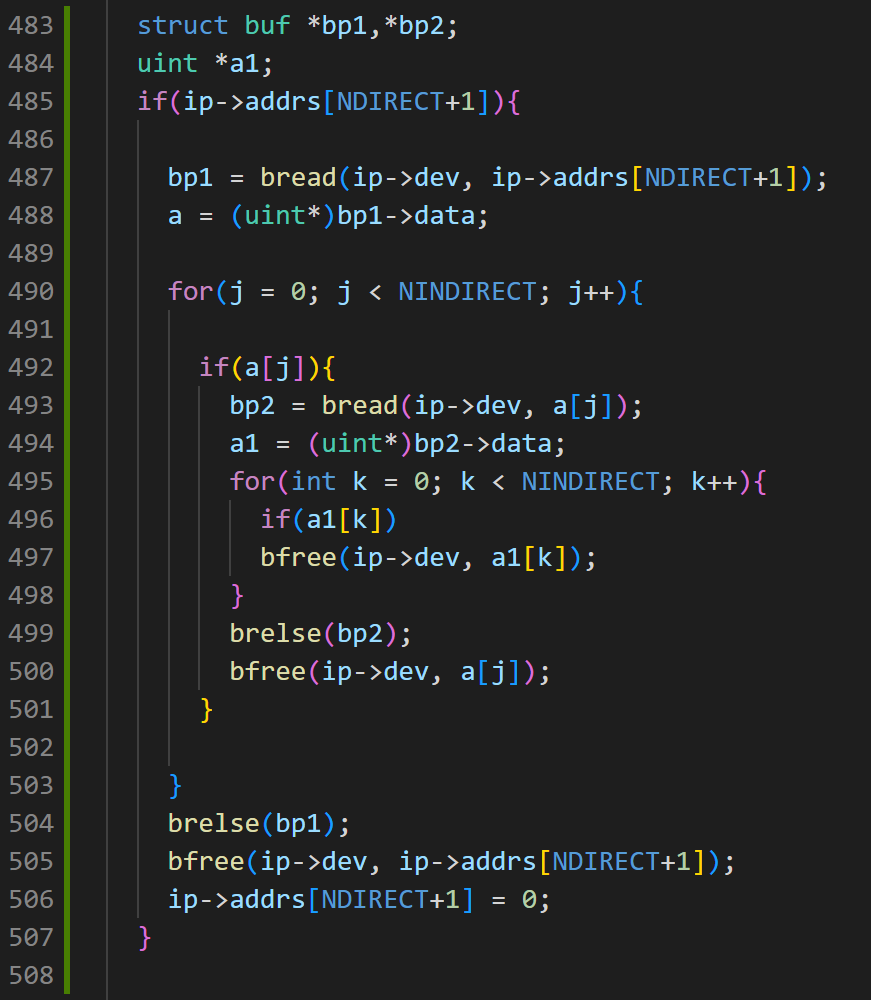
参照前面的写法写。

Bread用于将一个磁盘块缓存到一块内存块中。然后我们就对应修改内存中的副本就可以了。

目录中的每个entry都是4字节的块号，所以读取时将指针转化为uint\*就可以将整块目录转化为一个目录数组，就可以按entry进行访问了。

修改好的缓存要写回，不用的缓存要释放。

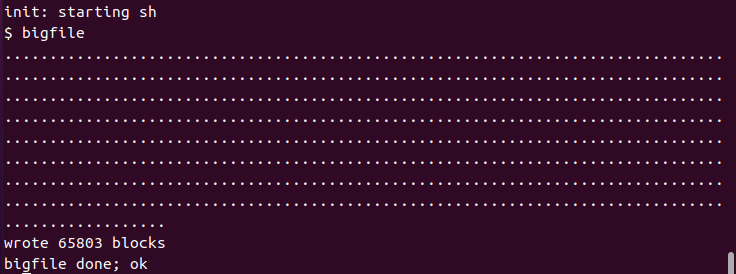
* 



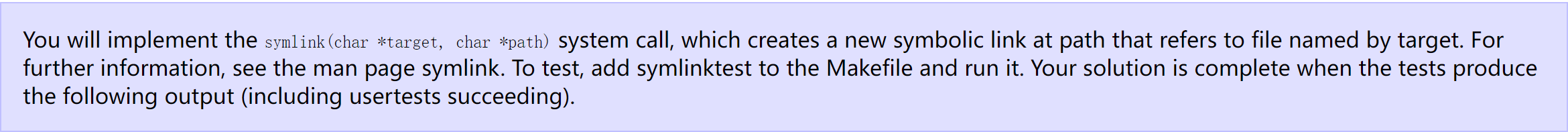
释放也是参照上面释放一级目录的写法就可以了。

测试结果截图：

运行成功



## **Symbolic links**



按照提示进行：

* 

对应添加：





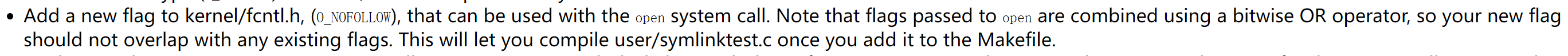






* 



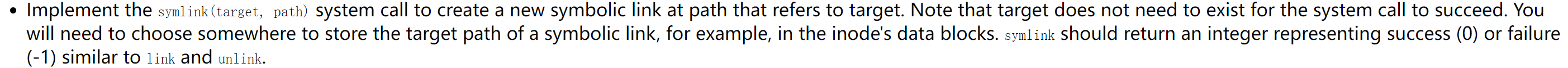
* 



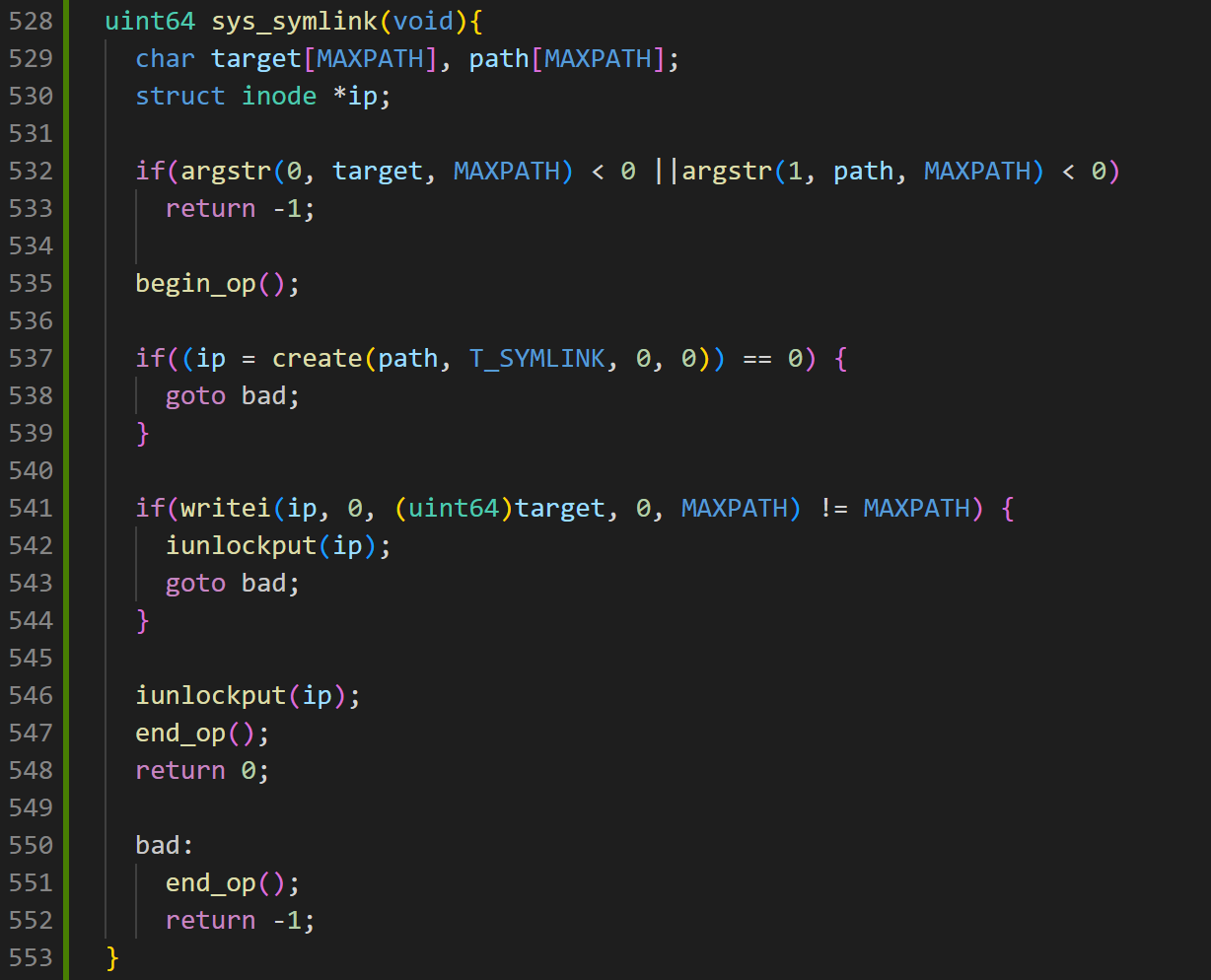
空其实很多的，随便挑了一个。

然后就可以修改Makefile了。



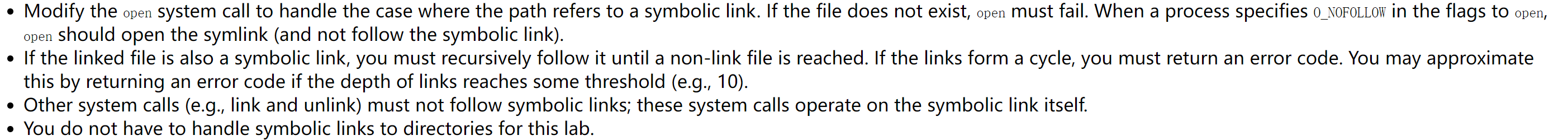
* 

直接在inode里保存target地址实际上就比较简单了。当然也可以单独再新建一个文件，在它里面存储地址，不过这样写起来就麻烦了，还会浪费空间。考虑xv6的文件名最大长度其实很短，存在inode里其实还是很合理的。对应就有：

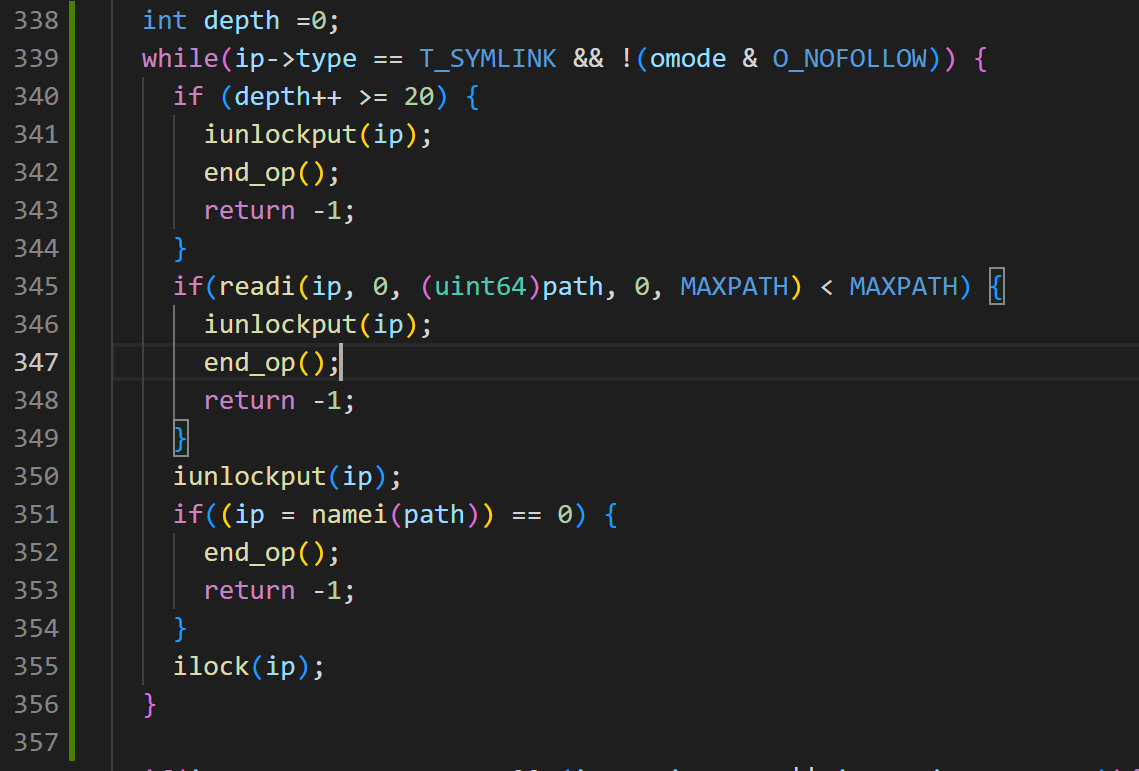


在path路径下创建一个symlink的inode，其中只存储路径，而非一般文件inode中的文件标识符或者目录。这个inode指示的文件会有独特的解析方式，这会在之后的open中实现。

创建修改完成后就可以了。主要问题还是在打开上。

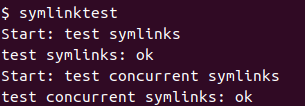
* 

对应修改sys\_open:



参照其他打开方式打开其他类型的文件的写法就可以了。虽然逻辑上说打开符号链接应该是递归实现，但因为实际上这是一个尾递归，所以没有必要非得用递归写法，循环就可以解决问题了。只需要一只循环直到找到不是符号链接的文件就可以了。或者深度太大就报错，因为可能是在绕圈。

测试结果截图：



# 问题和解决

## 调试和检查

在第一个实验中报了这样一个错：



检查发现似乎是因为inode大小不一致。

但是我明明修改过对应文件了呀？

实际上是因为保存的ctrl+S只保存了当前页，其他页没有保存。对应保存就都正确。

## XV6的文件系统和我使用的（Win）有什么异同？

应该说，差异主要体现在一些具体细节上，比如：

更小的文件大小、更短的最长文件名、更低的性能等等。

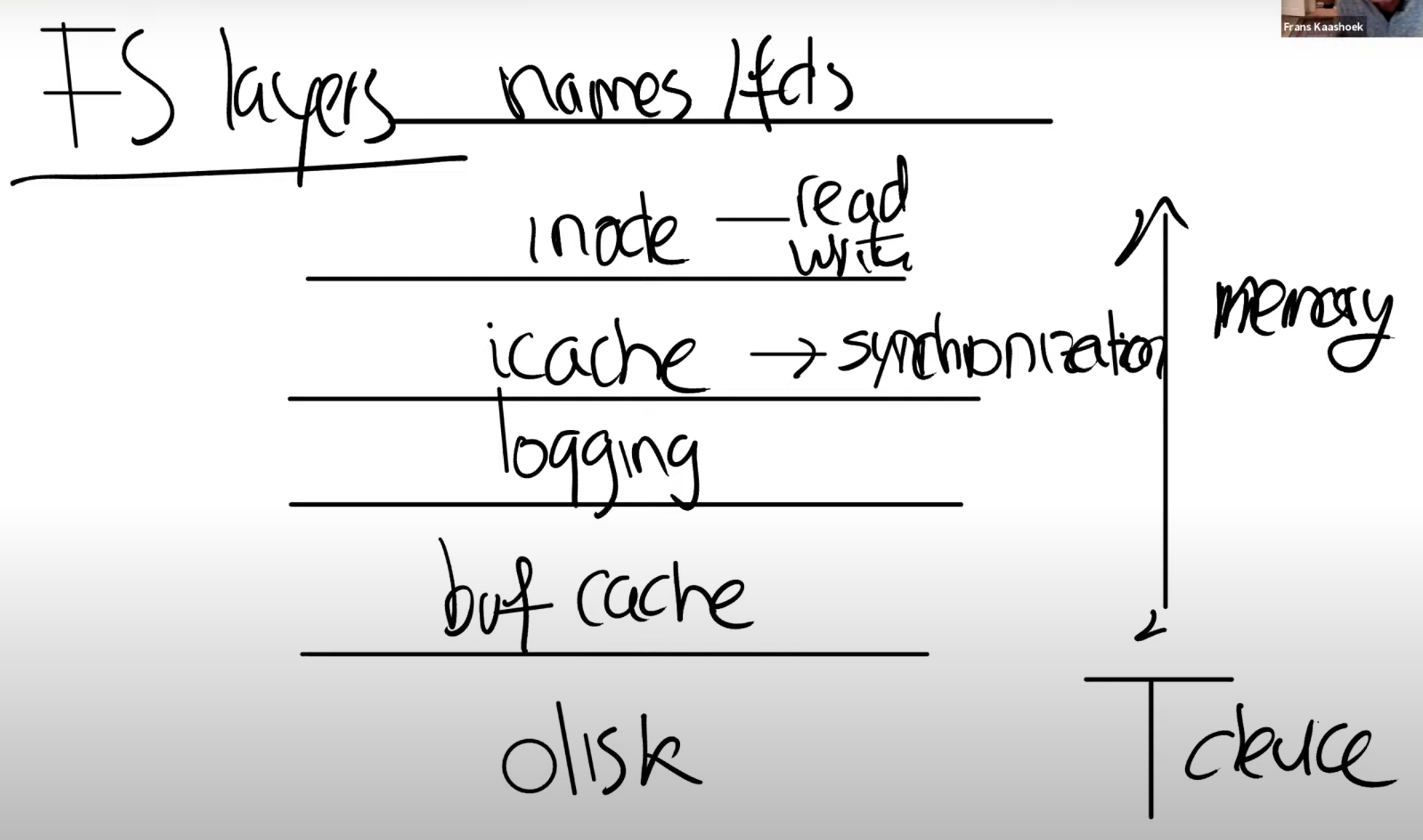
相同的部分实际上更多，他们都实现了文件系统应该实现的目的，即：

* 对于用户友好的文件名，具体来说就是层级的路径名，这可以帮助用户组织目录中的文件。
* 通过将文件命名成方便易记的名字，可以在用户之间和进程之间更简单的共享文件。
* 文件系统提供了持久化。

# 实验总结

## 简述XV6的文件系统

大概可以概括为这样一张图：



* 在最底层是磁盘，也就是一些实际保存数据的存储设备，正是这些设备提供了持久化存储。
* 在这之上是buffer cache或者说block cache，这些cache可以避免频繁的读写磁盘。这里我们将磁盘中的数据保存在了内存中。
* 为了保证持久性，再往上通常会有一个logging层。许多文件系统都有某种形式的logging。
* 在logging层之上，XV6有inode cache，这主要是为了同步。inode通常小于一个disk block，所以多个inode通常会打包存储在一个disk block中。为了向单个inode提供同步操作，XV6维护了inode cache。
* 再往上就是inode本身了。它实现了read/write。
* 再往上，就是文件名，和文件描述符操作。