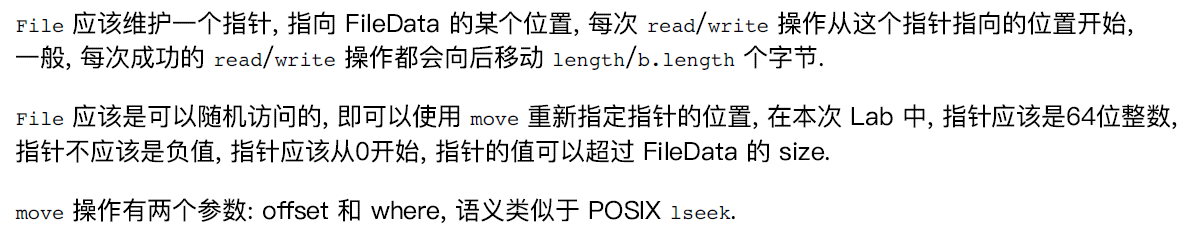
本次的lab1是关于实现AlphaFileSystem。我只是根据自己的实现思路提供以下提示。

①需要一个专门进行字符串处理的类。本次lab基于已有的计算机文件系统。由于block manager和file 都涉及对于文件的读写操作。问题的复杂性在于，助教给的filemeta文件格式需要我们在读写的时候对保存在其中的字符串元数据进行修改。因此建议有一个专门的字符串处理类来处理相关内容。例如：对包含block manager id和block indexId的字符串进行裁剪，获取文件路径名。

②推荐从block和block manager开始实现。block类的实现相对更简单，而且实现的过程中可以帮助我们理清整个系统应有的实现逻辑。

③难点在于File实现类的read()和write()函数。首先，注意看lab文档。



File维护的指针会影响读写的位置。因而，读写的时候需要进行的操作要比想象的复杂。如果根据助教在群中提到的“可以不实现文件空洞”的实现方案，思路就比较清晰了。

以write()操作为例。write()函数首先找到指针（就像我们平常写文档时候出现的光标）位置。这个位置有两种情况：

1. 在已有数据的内部。我们需要利用字符串处理函数从filemeta文件中找到该位置之前数据所在的block列表。而这又有两种情况
2. 指针位置恰好在某一个block内容的尾部。去除filemeta中指针后的block indexid等信息。创建新的block并写入数据，同时修改filemeta。
3. 指针位置在某一个block内容的中部。首先复制该block内容（blockdata）中指针前的部分，创建新的byte[]数组拼接要写入的内容再进行类似1）的过程。
4. 在已有数据的尾部。直接创建新的block，并写入数据，修改blockmeta和filemeta即可

④Duplication的实现：

可以从已有的block manager列表中随机选择n个各自创建block并写入data，修改meta。这个n可以写死也可以用户输入。