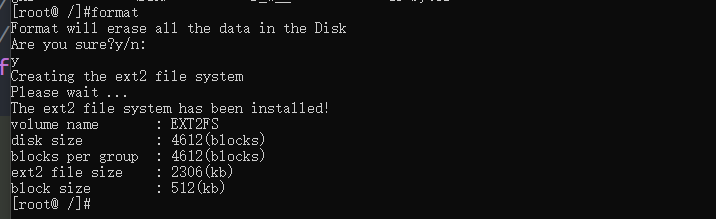
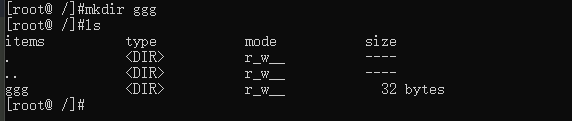
第三次实验：**类EXT2文件系统的设计**

**1.测试程序功能：**

**1.格式化**



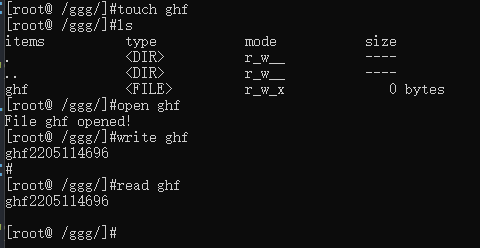
**2.创建目录并查看**



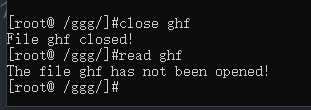
**3.** 进入所创建的目录创建文件并查看



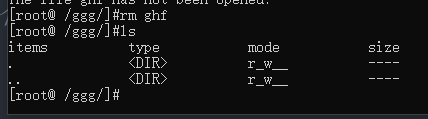
**4.打开文件并且输入内容，然后读取该文件**



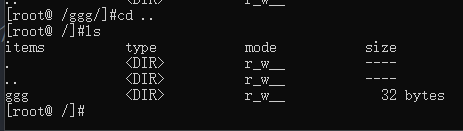
**5.关闭文件后读取该文件**



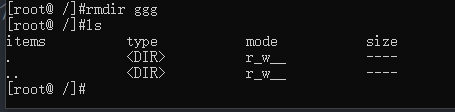
**6.删除该文件**



**7.返回上级目录**



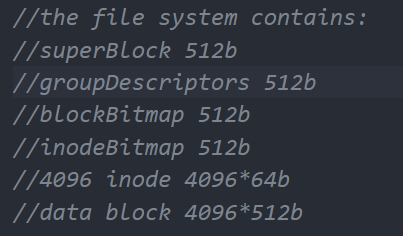
**8.删除目录**



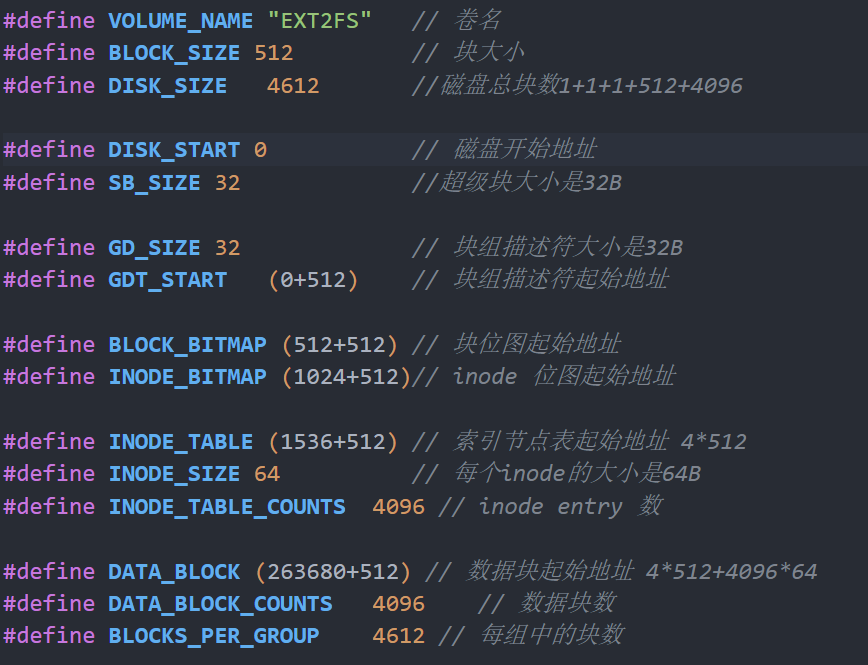
**9.退出文件系统**

C:\Users\lenovo\Documents\Tencent Files\1162984920\Image\Group2\2E\1E\2E1E@L8G0HL6CV3HHQH7`AI.png

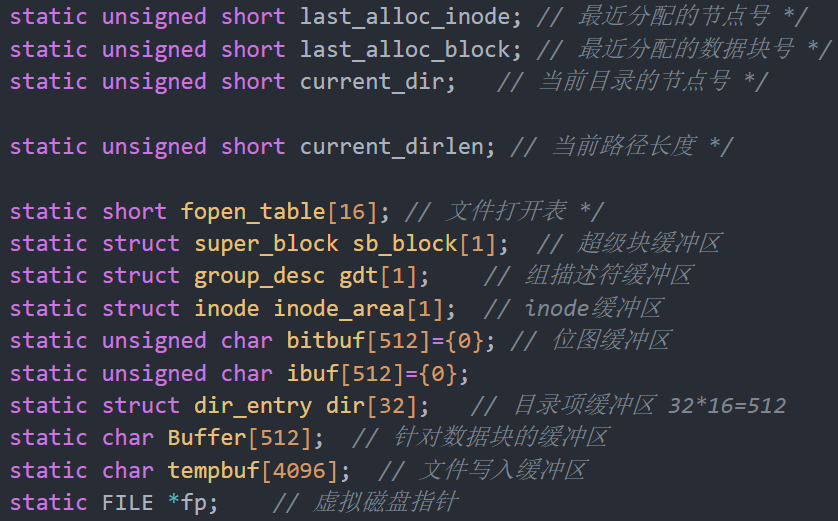
**2.文件系统的组织结构（逻辑组织结构）：**



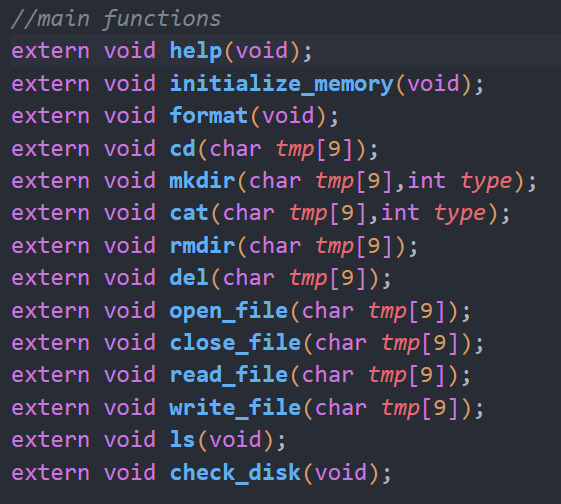
设计的一些常量：



进行超级块、组描述符、位图、inode 节点、数据块的读写时，需要用到一些缓冲区，定义如下：

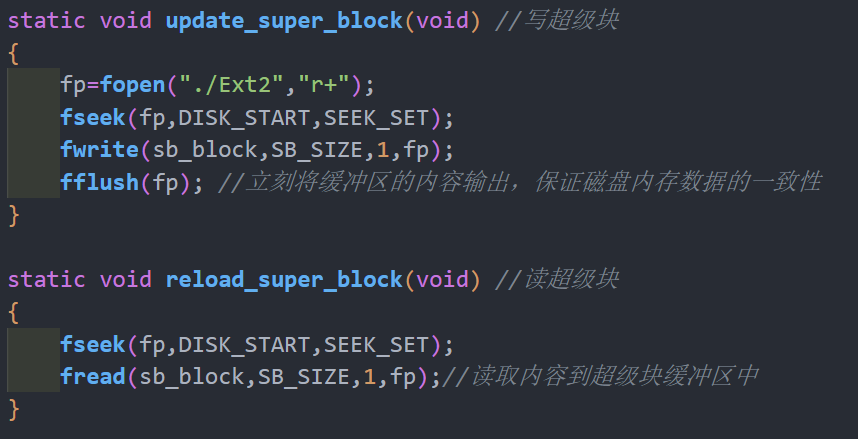


实现的主要功能：



**3.函数说明：**

1.读写函数：



使用了fseek 和fwrite函数，fseek（文件指针，偏移量，文件开头位置）

Fwrite(要输入内容，输入内容大小，单个数据大小，文件指针)fread函数类似。写完后再用fflush函数清除读写缓冲区

类似的读写函数：



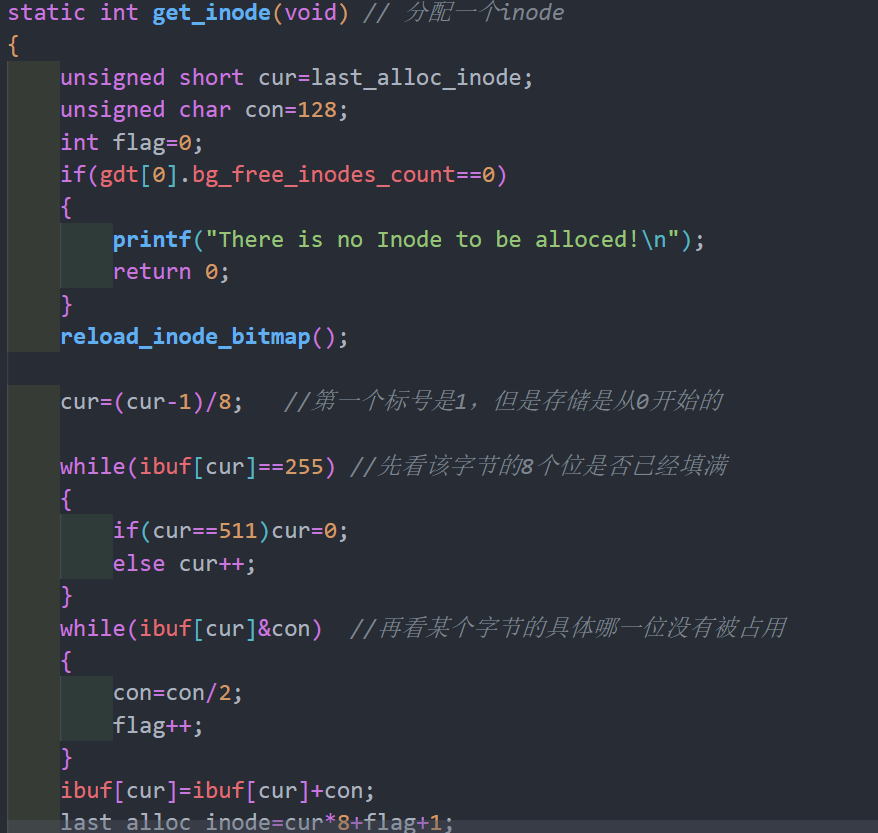
2.分配数据块：

先找在哪个字节，再找字节哪一段，最后分配数据块



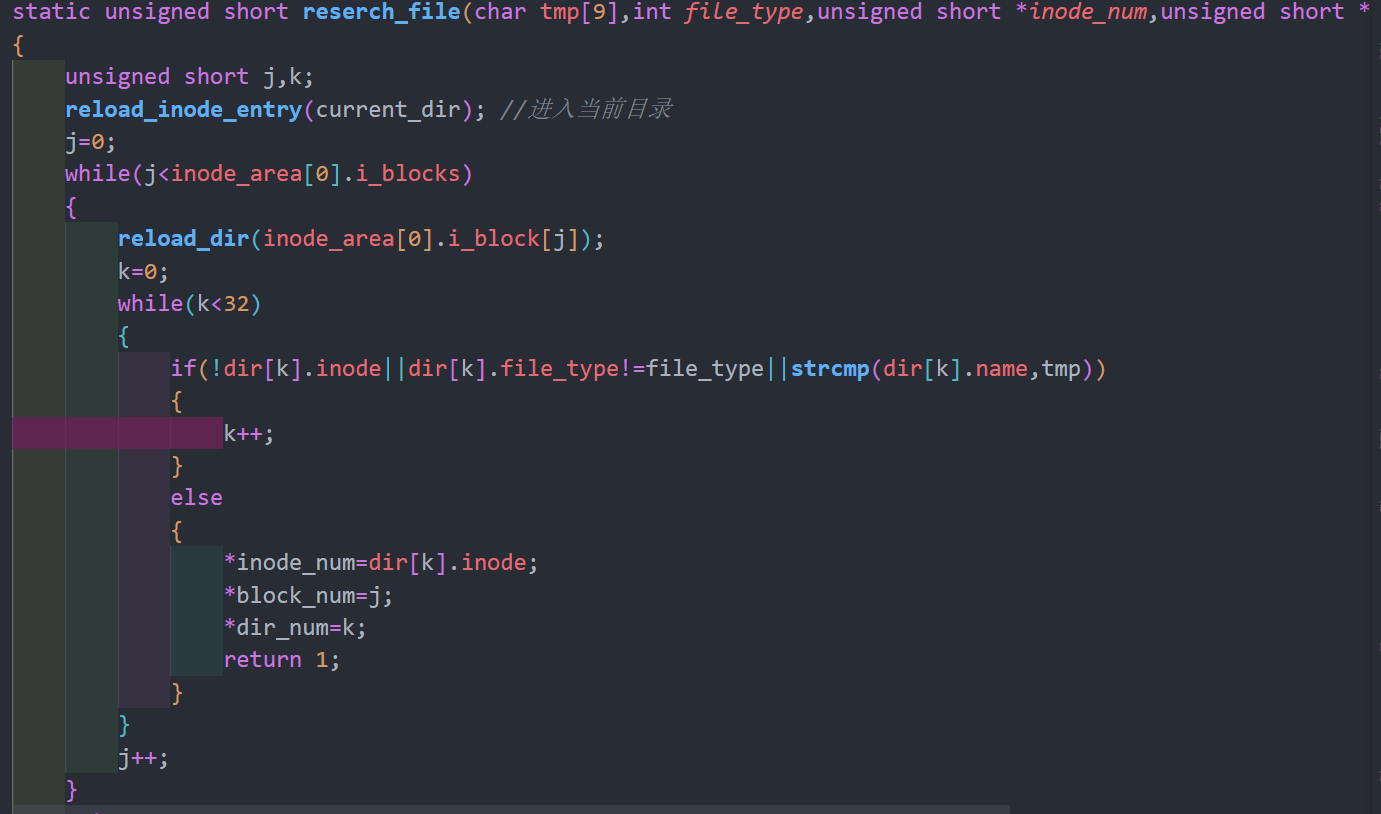


3.indode分配，思路类似，但是由于inode标号是从0开始，故cur-1：

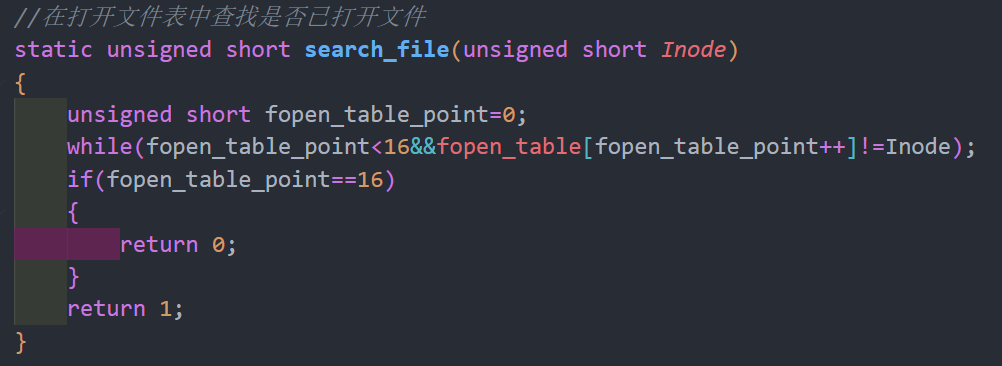


4.查找文件：

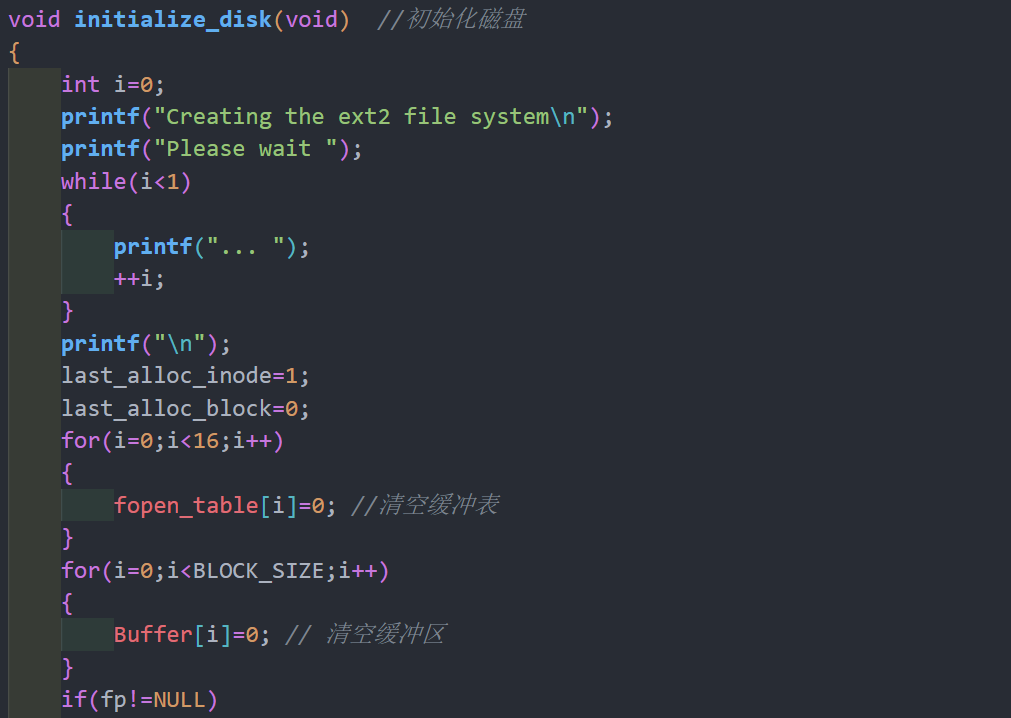
进入当前目录，外循环为i\_blocks数，内循环是32，因为dir缓冲区一次只能读32个，查找成功返回1，失败返回0；block\_num为J,dir\_num为K



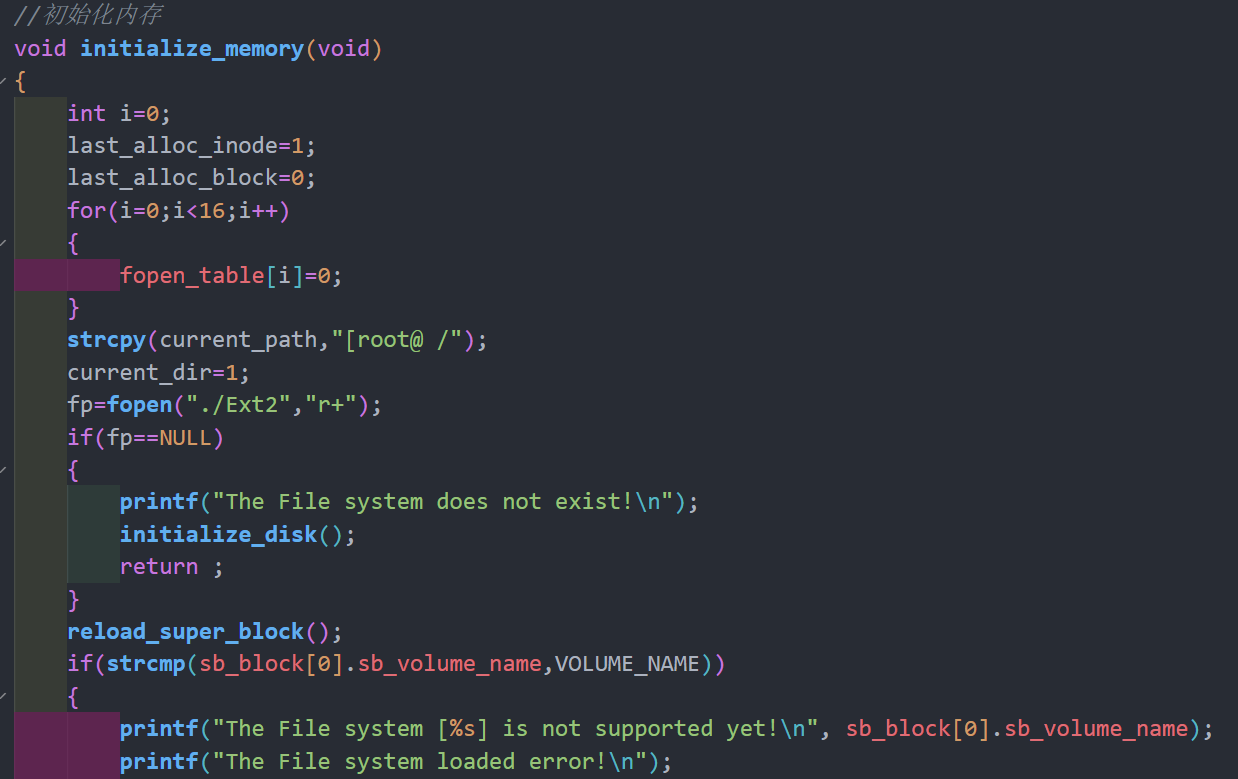
5.查找是否已经打开了该文件，找到返回1，没有则0



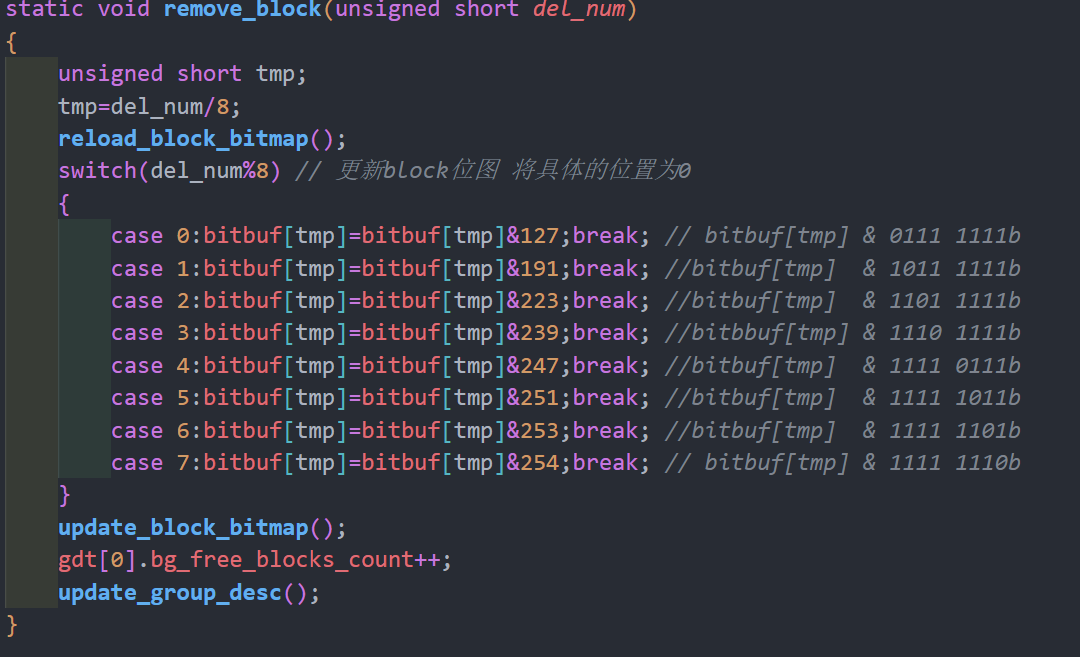
6.初始化磁盘：



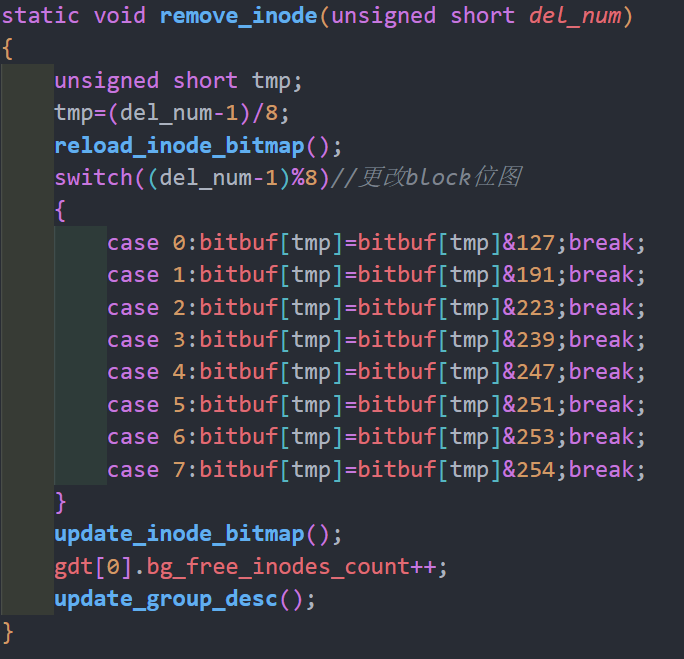
7.初始化内存：



8.删除块：

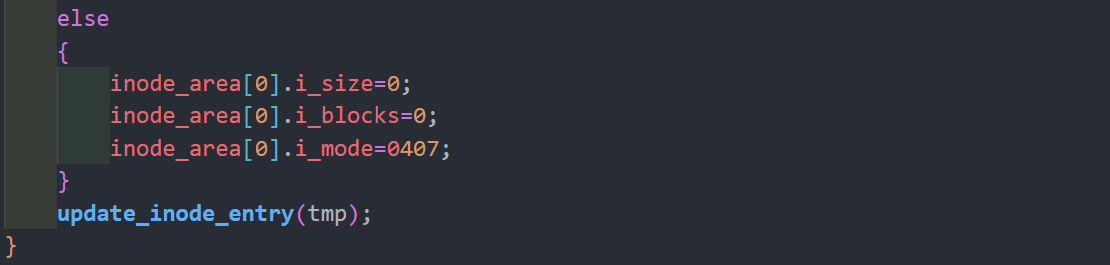


10.删除inode：与删除block类似的操作



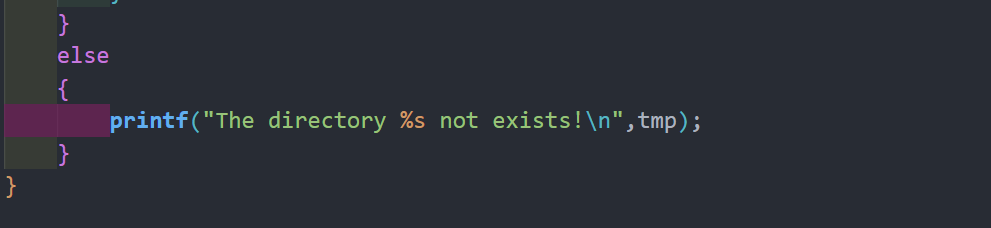
11.为新增目录或者文件分配dir entry



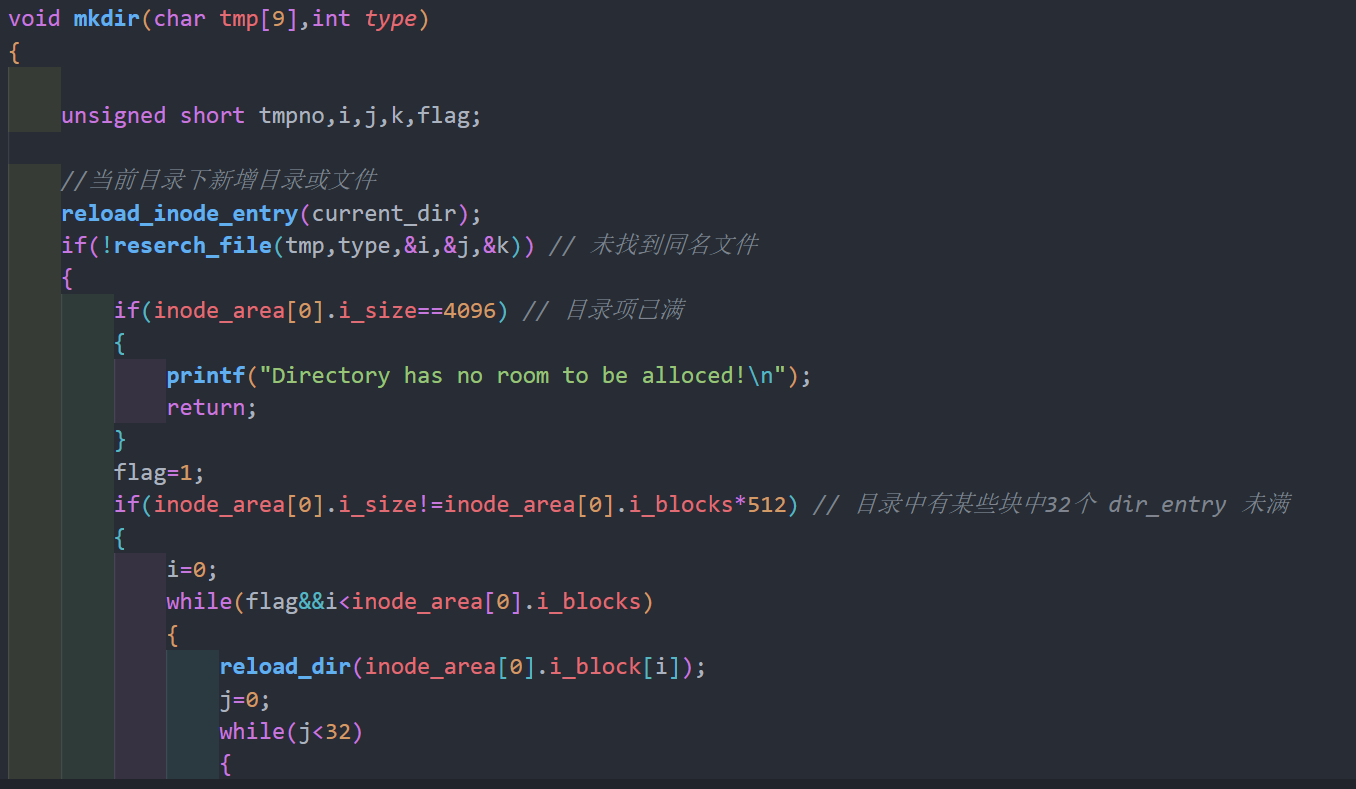


12.cd：进入某个文件夹内：





13.创建目录：



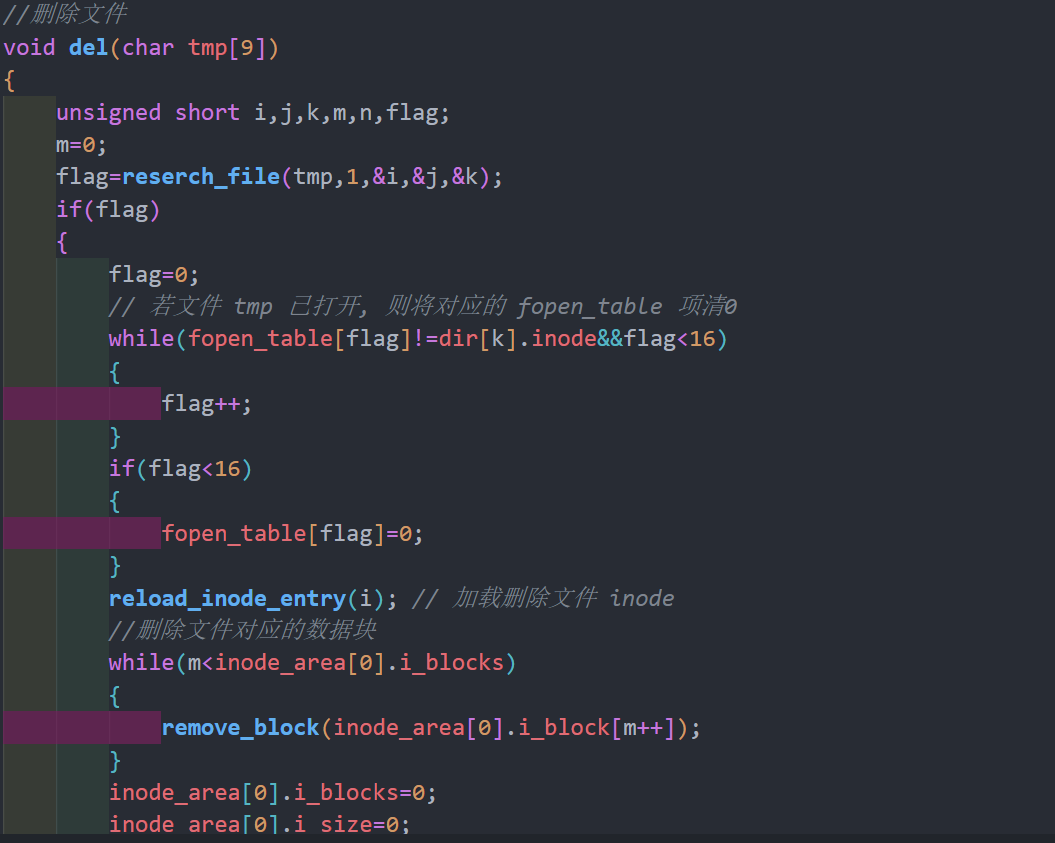
14.创建文件：



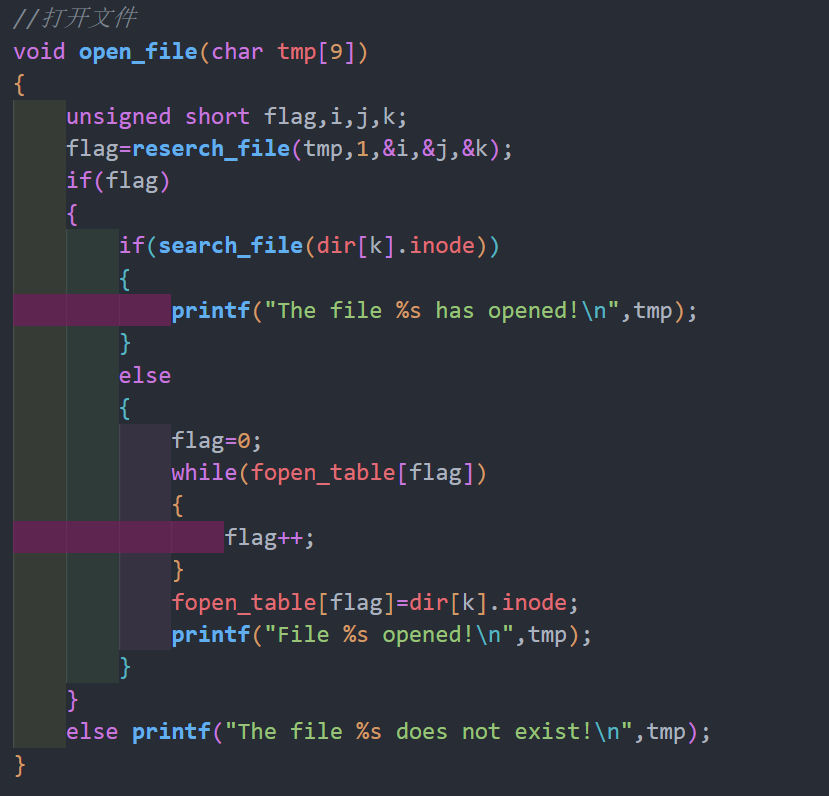
15.删除空目录：



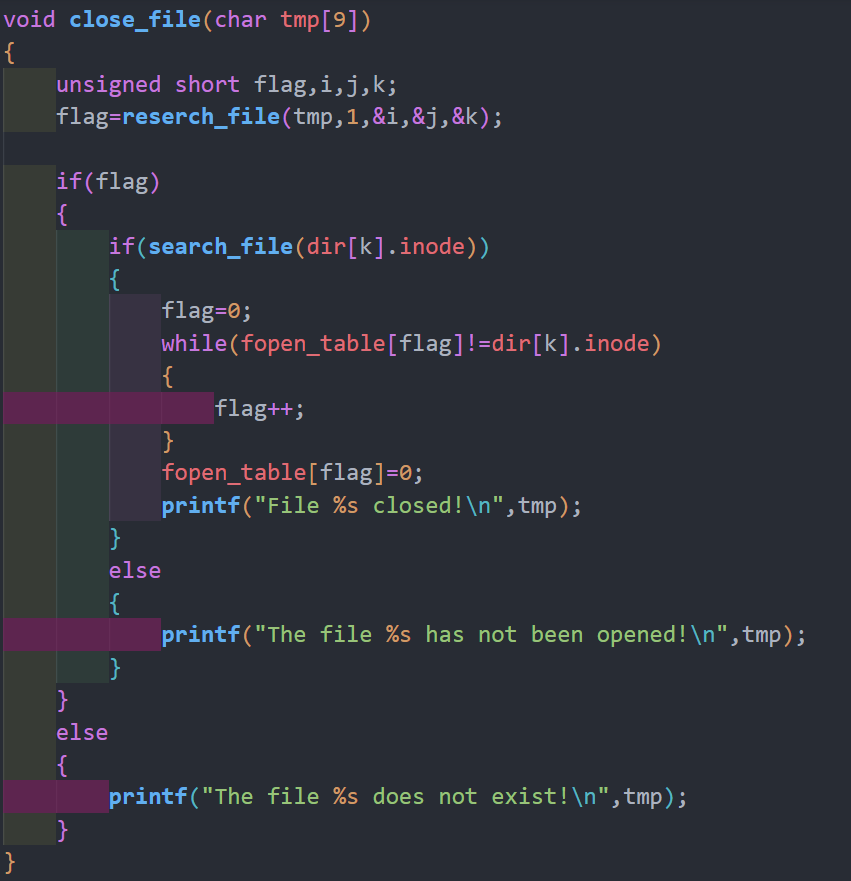
16.删除文件：



17.打开文件：



18.关闭文件：



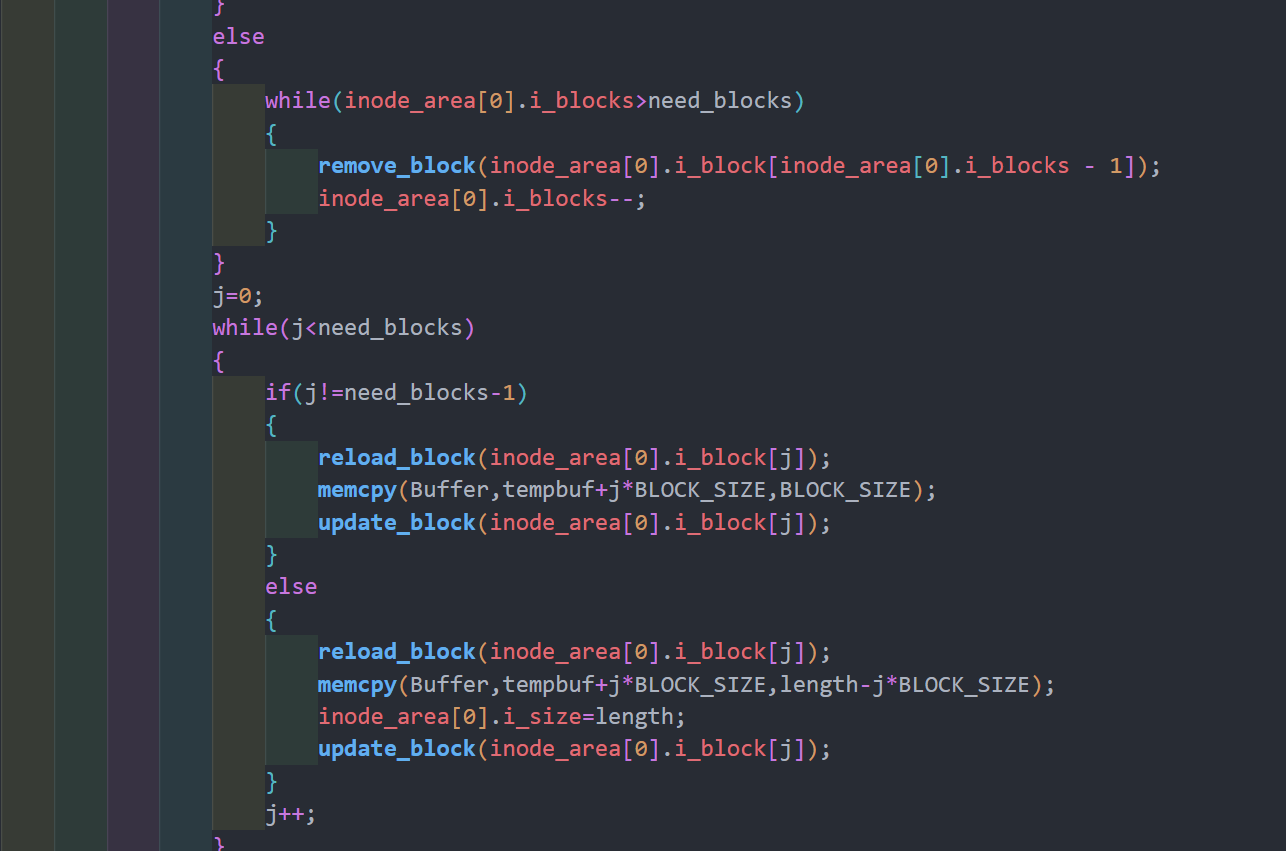
19.读入文件

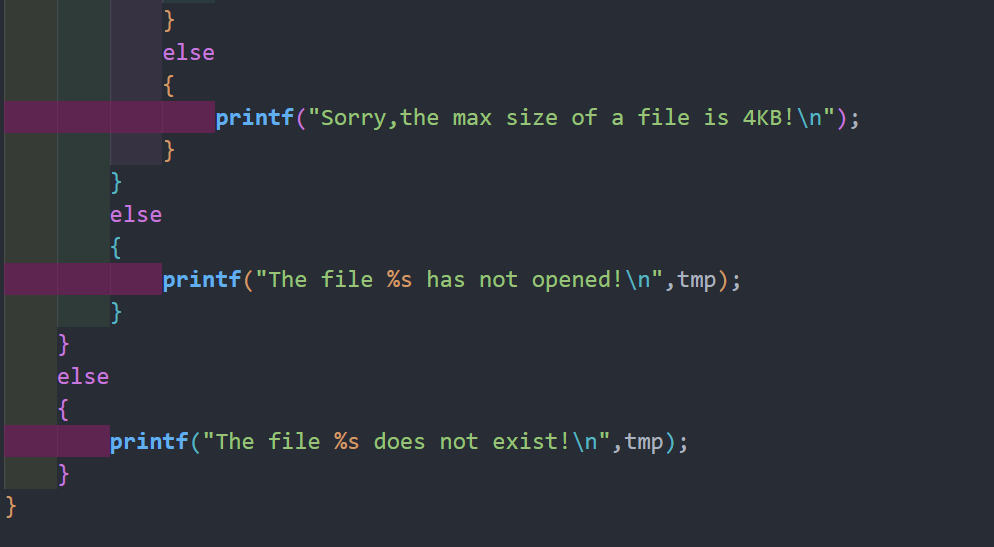


20.写入文件









21.ls函数

