**《计算机操作系统》实验报告**

**实验题目：请求页式存储管理**

**姓名：汪雨卿 学号：19120191 实验日期：2021.12.23**

**实验环境：**

VScode; C++

**实验目的：**

随着社会信息量的极大增长，要求计算机处理的信息与日俱增，涉及到社会生活的各个方面。因此，文件管理是操作系统的一个极为重要的组成部分。学生应独立地用高级语言编写和调试一个简单的文件系统，模拟文件管理的工作过程。从而对各种文件操作命令的实质内容和执行过程有比较深入的了解，掌握它们的实施方法，加深理解课堂上讲授过的知识。

**实验内容：**

（1） 本实习设计一个 10 个用户的文件系统，每个用户最多可保存 10 个文件，一次运行中用户可打开 5 个文件。

（2） 程序采用二级文件目录，即设置了主文件目录（MFD）和用户文件目录（UFD）。前者应包含文件主（即用户）及他们的目录区指针；后者应给出每个文件主占有的文件目录，即文件名，保护码，文件长度以及他们存放的位置等。另外为打开文件设置了运行文件目录（AFD），在文件打开时应填入打开文件号，本次打开保护码和读写指针等。

（3） 为了便于实现，对文件的读写作了简化，在执行读写命令时，只修改读写指针，并不进行实际文件的读写操作。

**操作过程：**

**1. 涉及的数据结构**

主文件目录和用户文件目录 如 图1 MFD和UFD的数据结构 所示

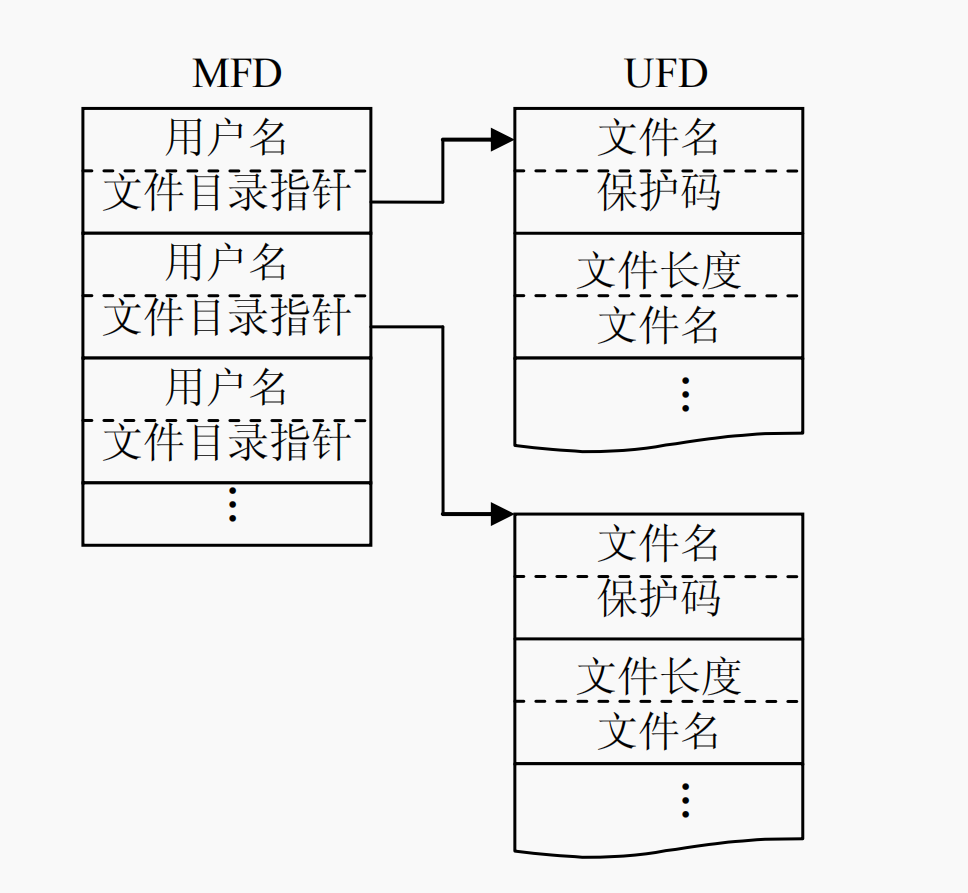


图1 MFD和UFD的数据结构

打开文件目录 如 图2 打开文件目录 所示

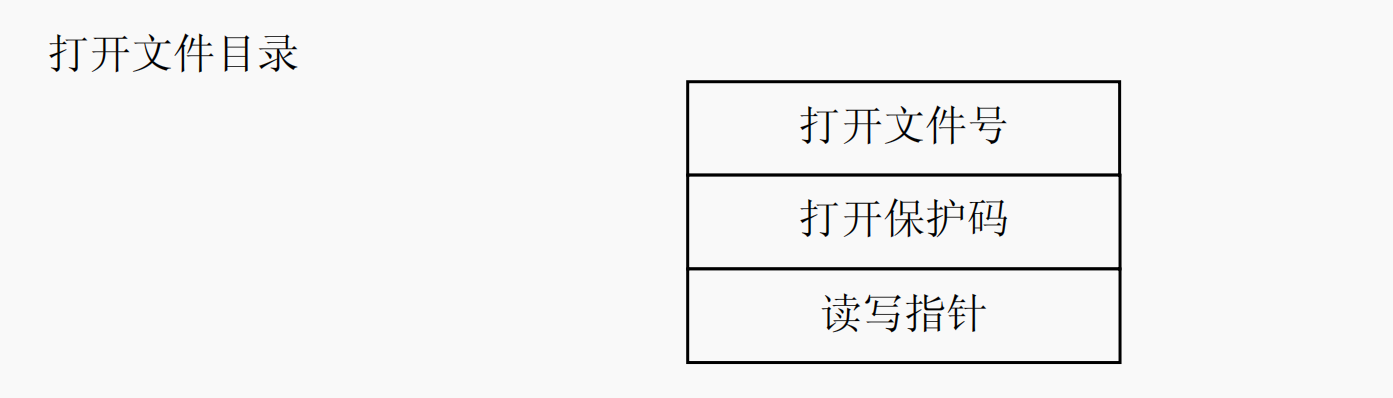


图2 打开文件目录

**2. 算法流程**

本程序实现的是链表加队列的数据结构形式来完成这个文件管理系统。首先，创建一个队列MFD，MFD上的每一个结点都代表了一个用户。每个用户拥有一个UDF队列， 对于UFD中的每一个结点，用户拥有对该文件进行增删改查的权利。其中，再创建一个文件队列，它作为打开文件队列，储存当前用户打开的文件列表。

流程介绍：首先进入系统选择用户，然后用户通过六个选项对文件进行增删改查，其中详情将在后面呈现。

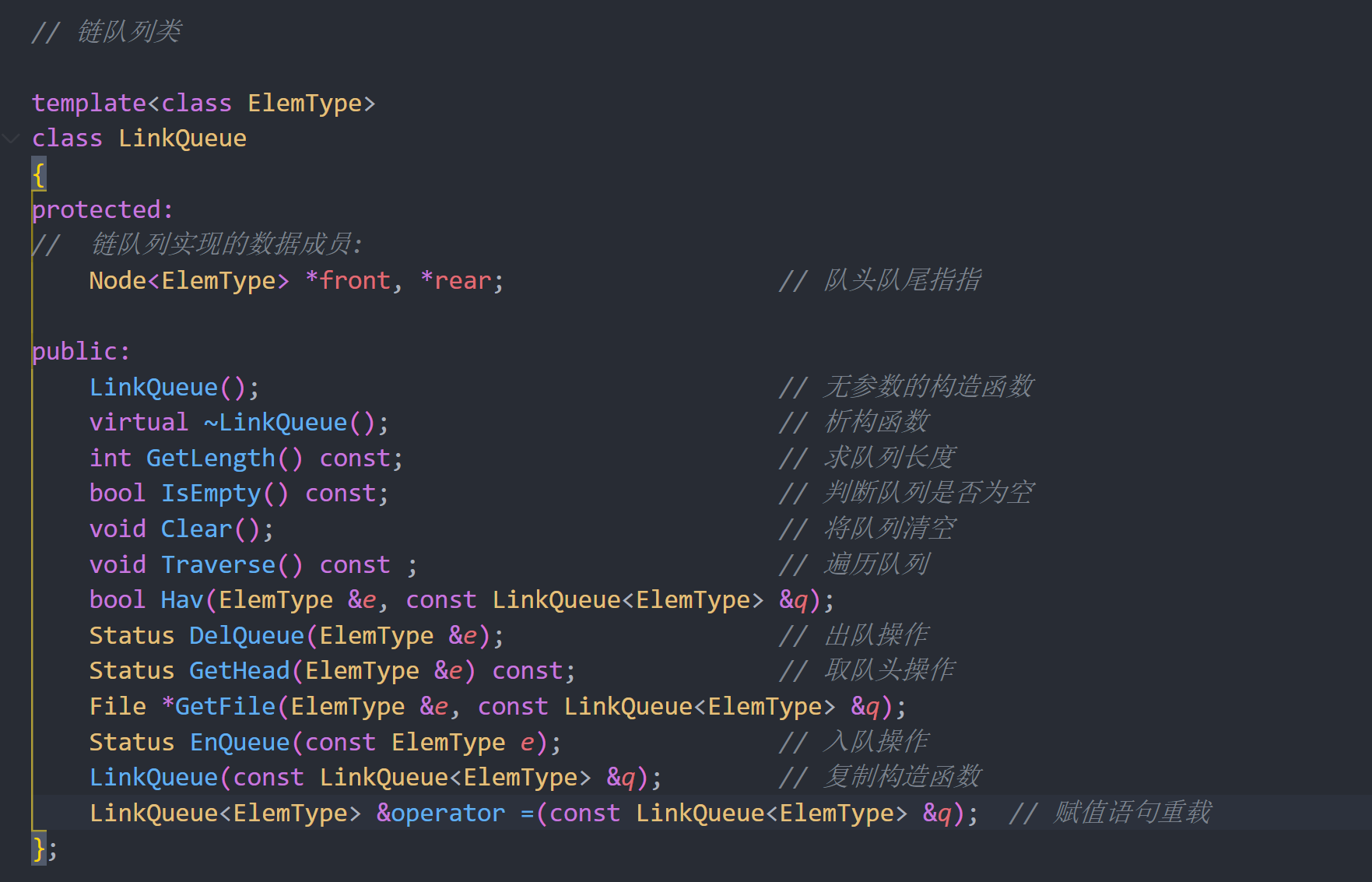


**3. Node.h 用户类 以及 文件类的实现**

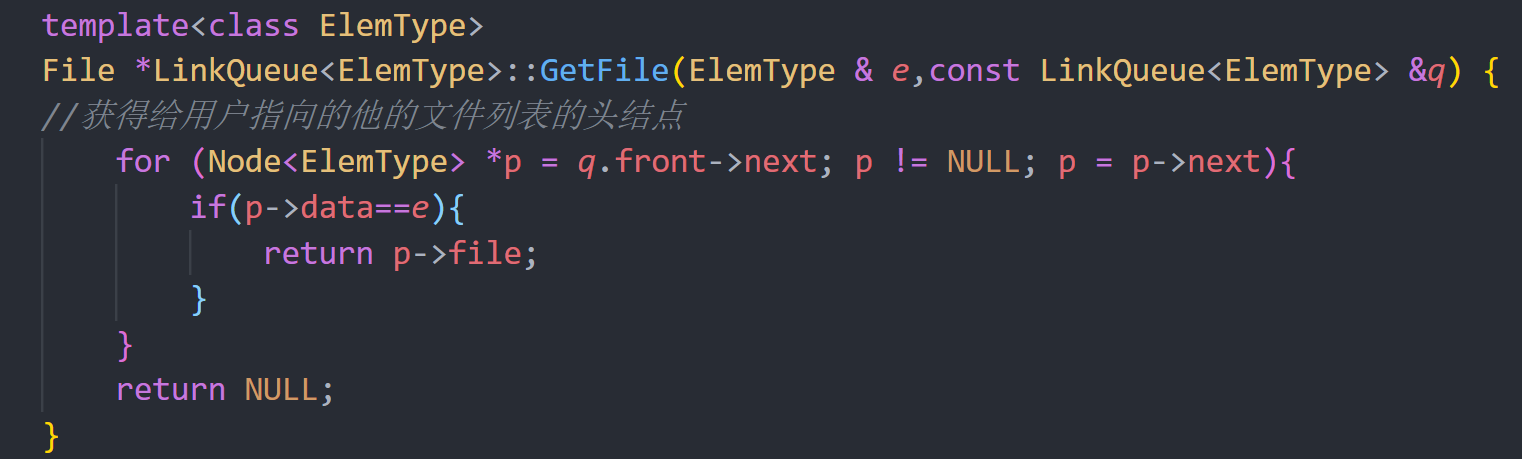
分别定义了用户类和文件类，并设置无参和带参数的构造函数



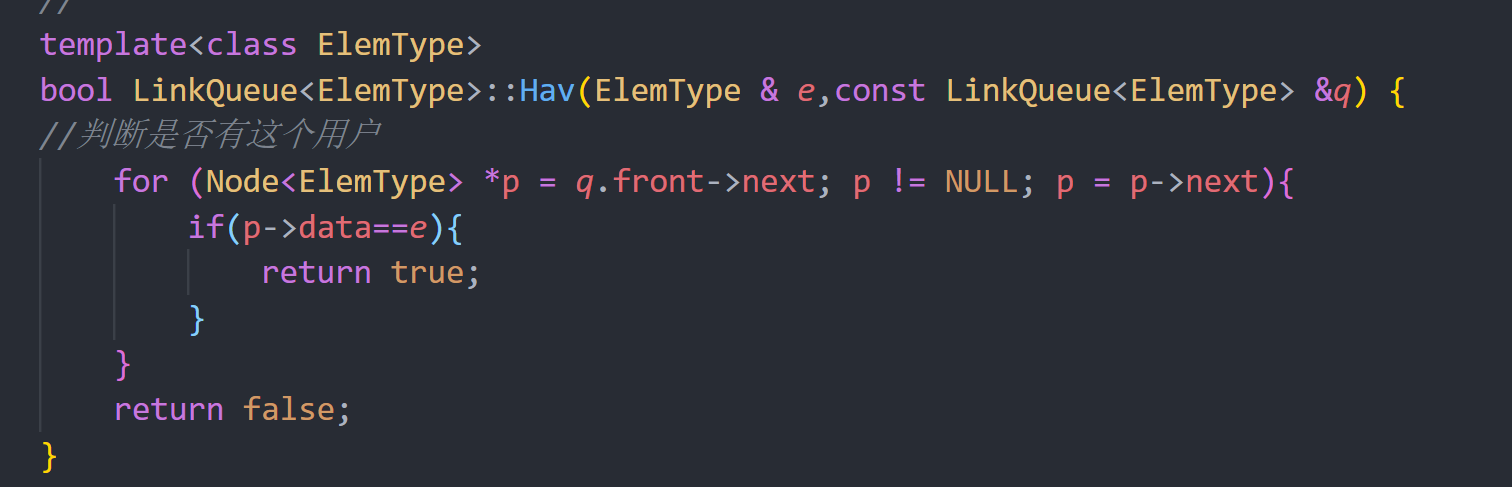
**4. LinkQueue.h中链表功能及实现（部分）**



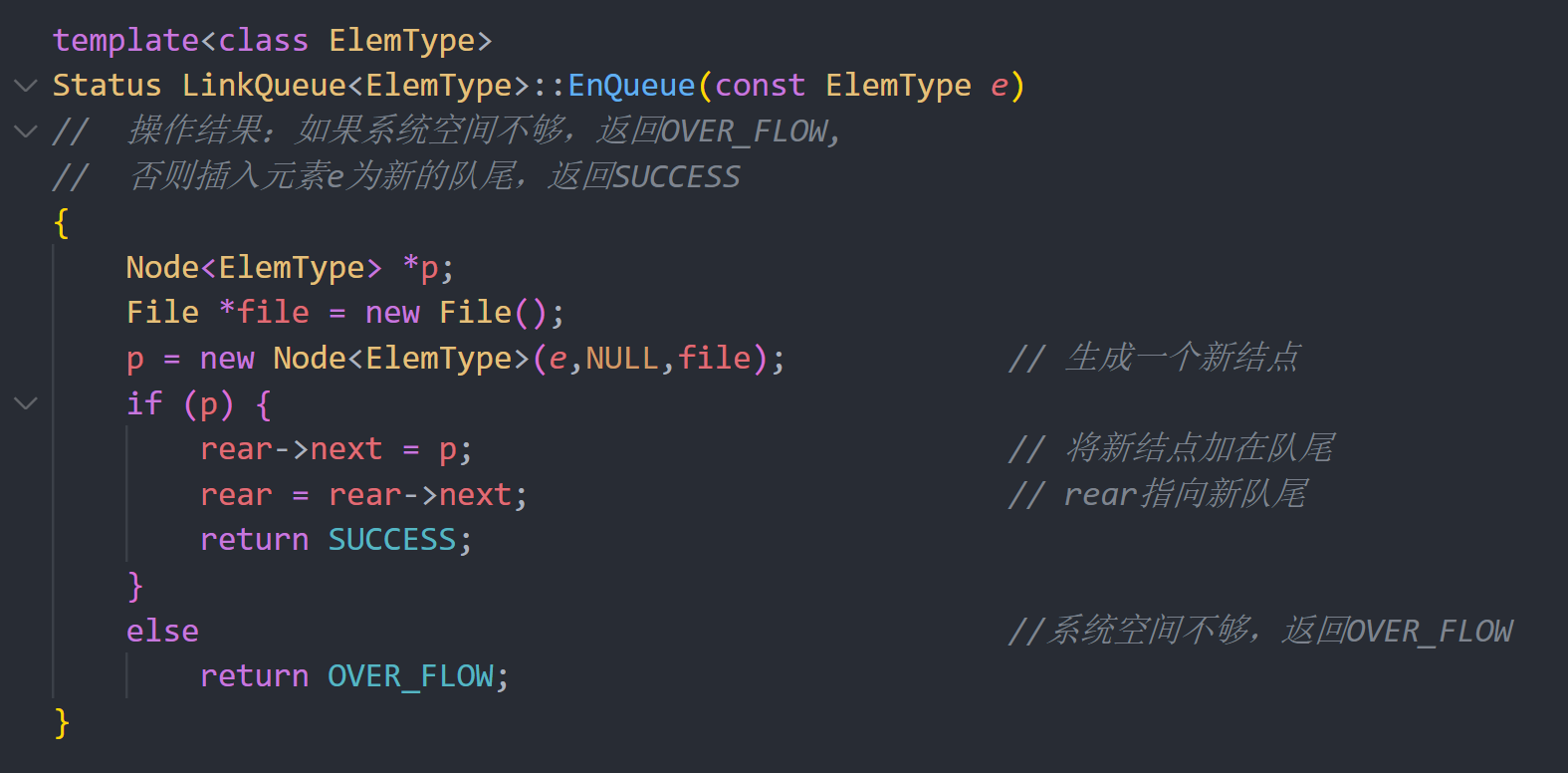
a. getFile():获取用户的文件列表的头结点



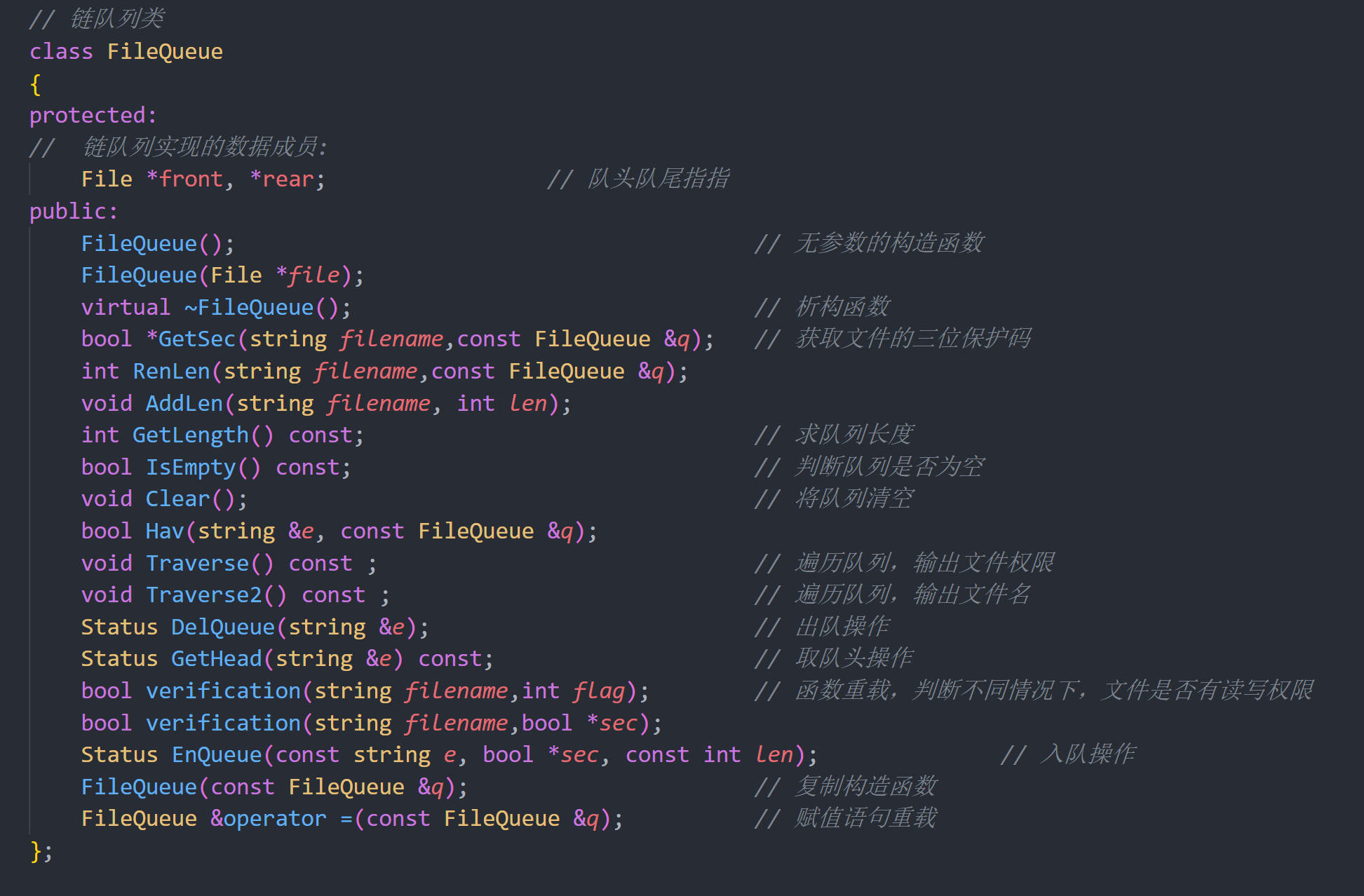
b. 判断用户队列是否有该用户



c. EnQueue(): 新增节点，把文件插入队列

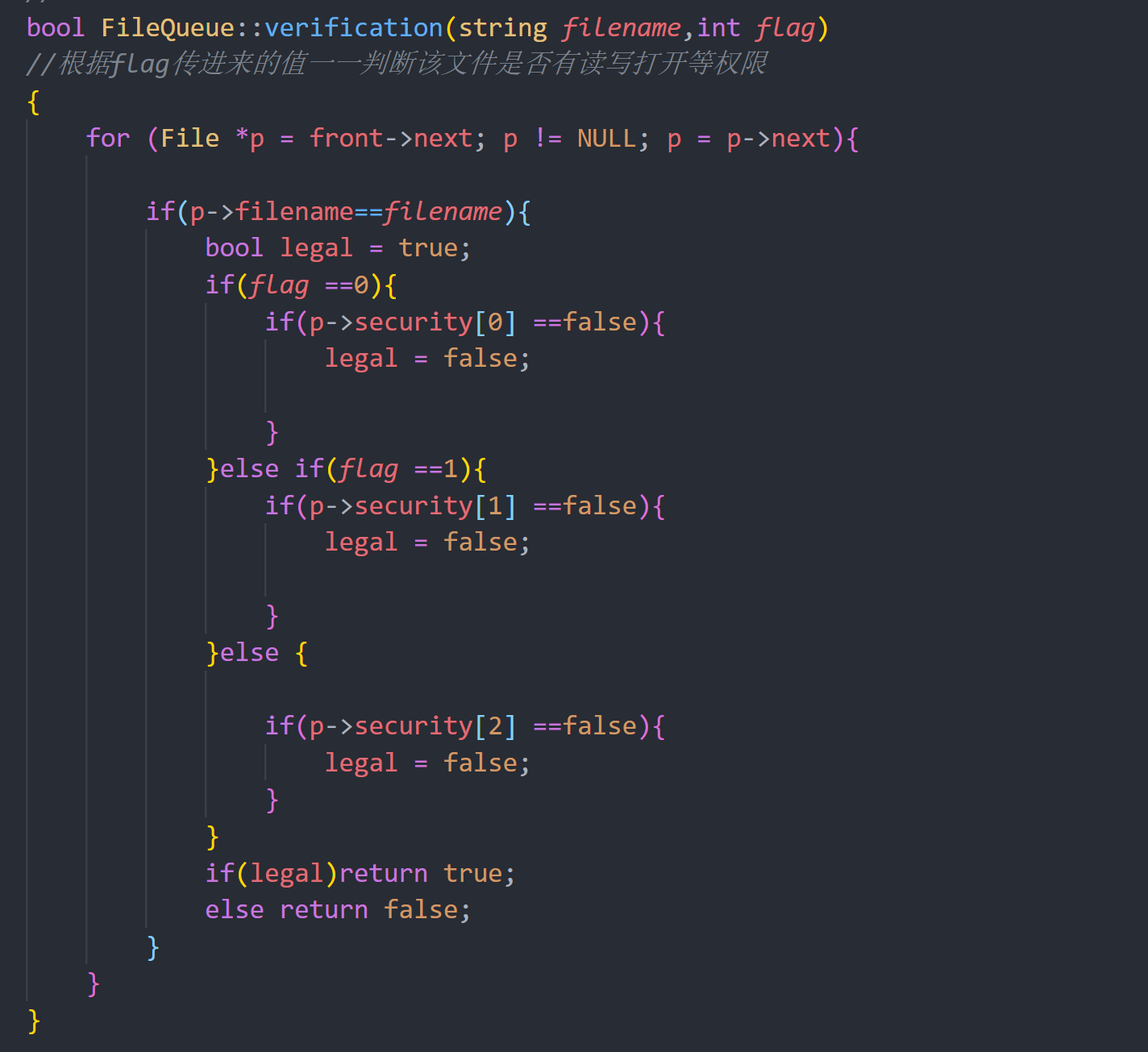


**4.FileQueue.h中功能及实现（部分）**

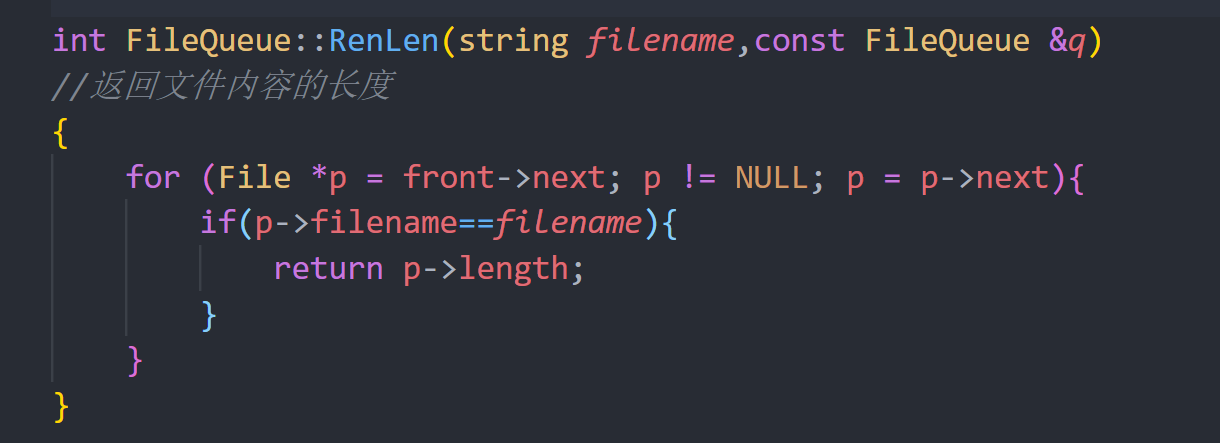


a. verification() 重载了两个方法：用于读取文件中的安全码验证。

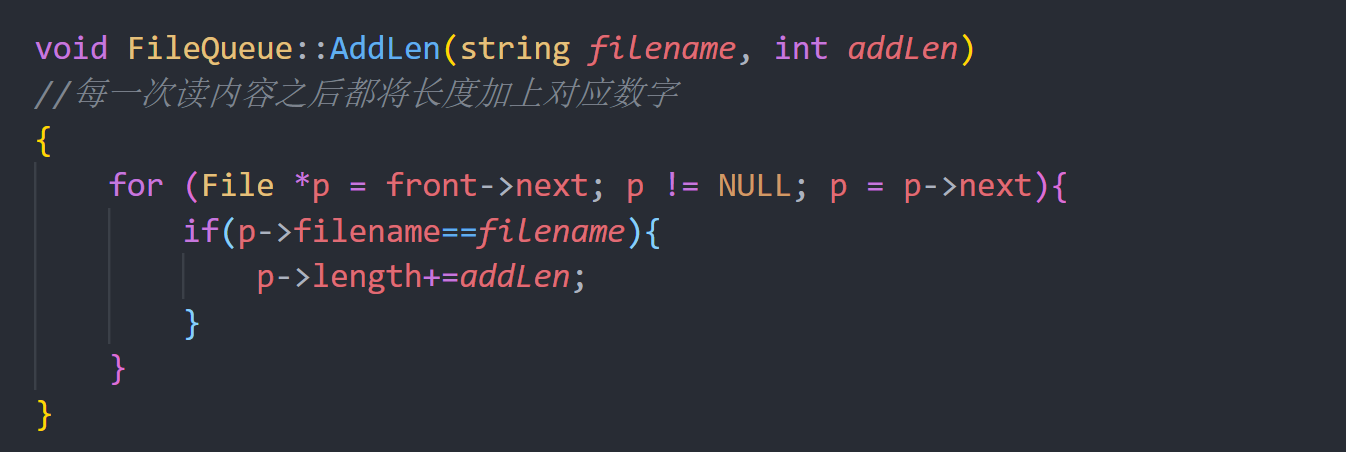




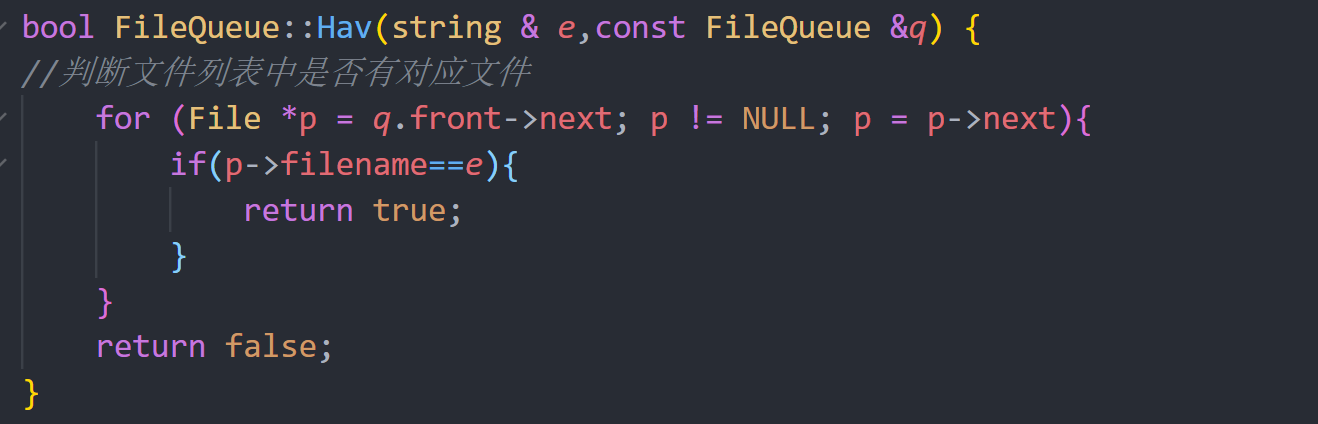
b. RenLen():返回文件内容的长度。



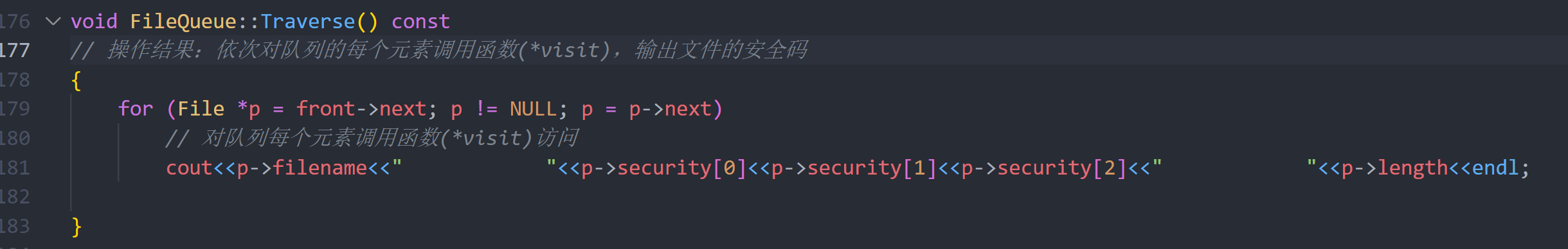
c. Addlen():添加增加的文件内容，重置文件长度。

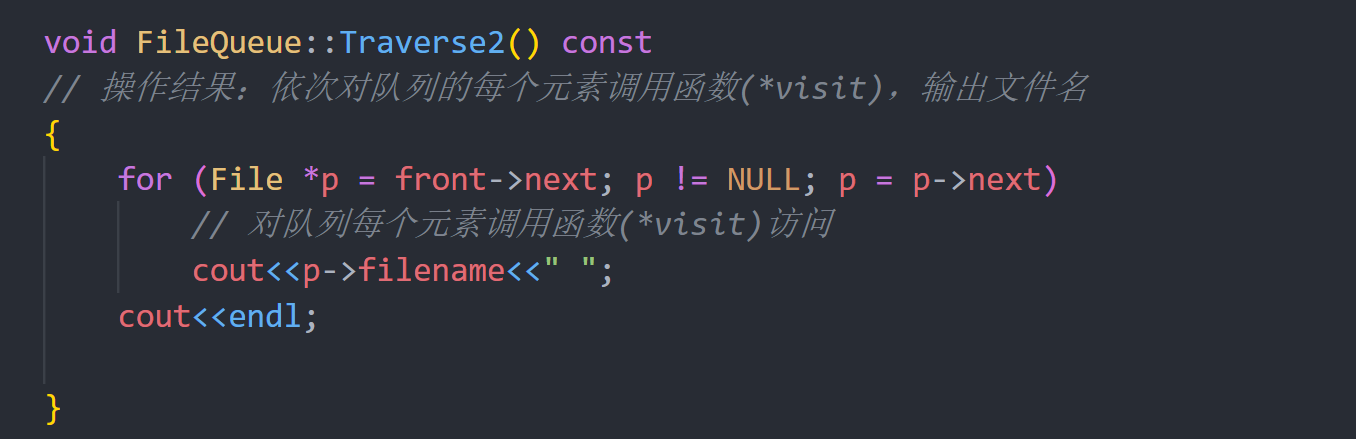


d. Hav():判断文件列表中是否有对应文件。



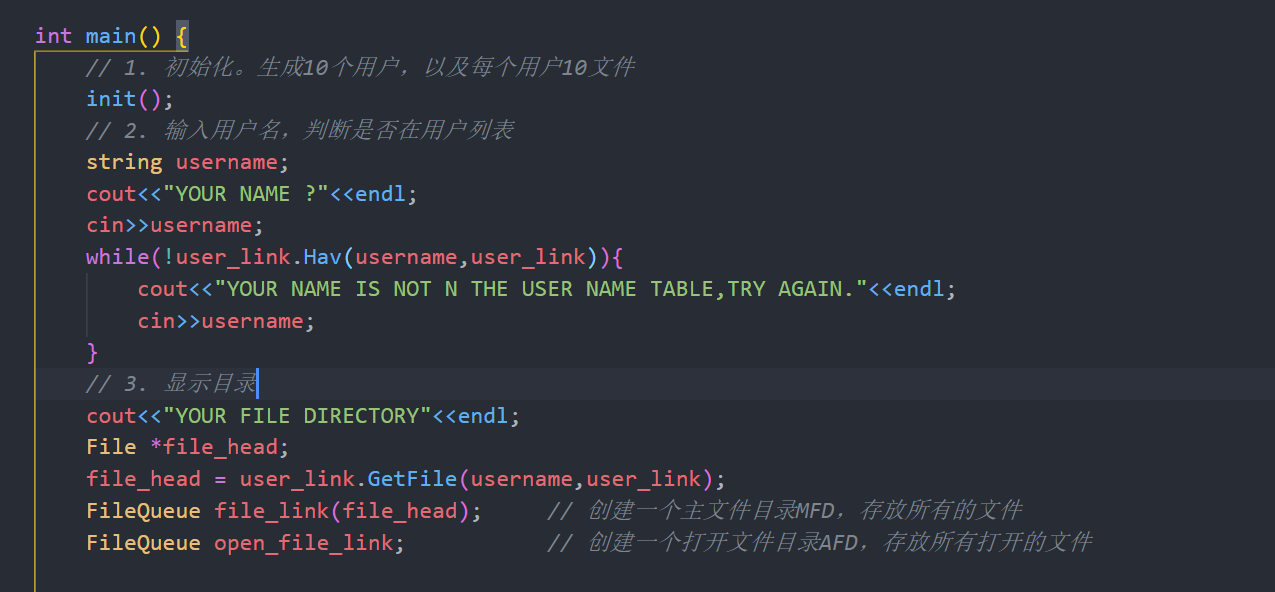
e. Traverse()，Traverse2():输出文件的信息：文件名、文件保护码。

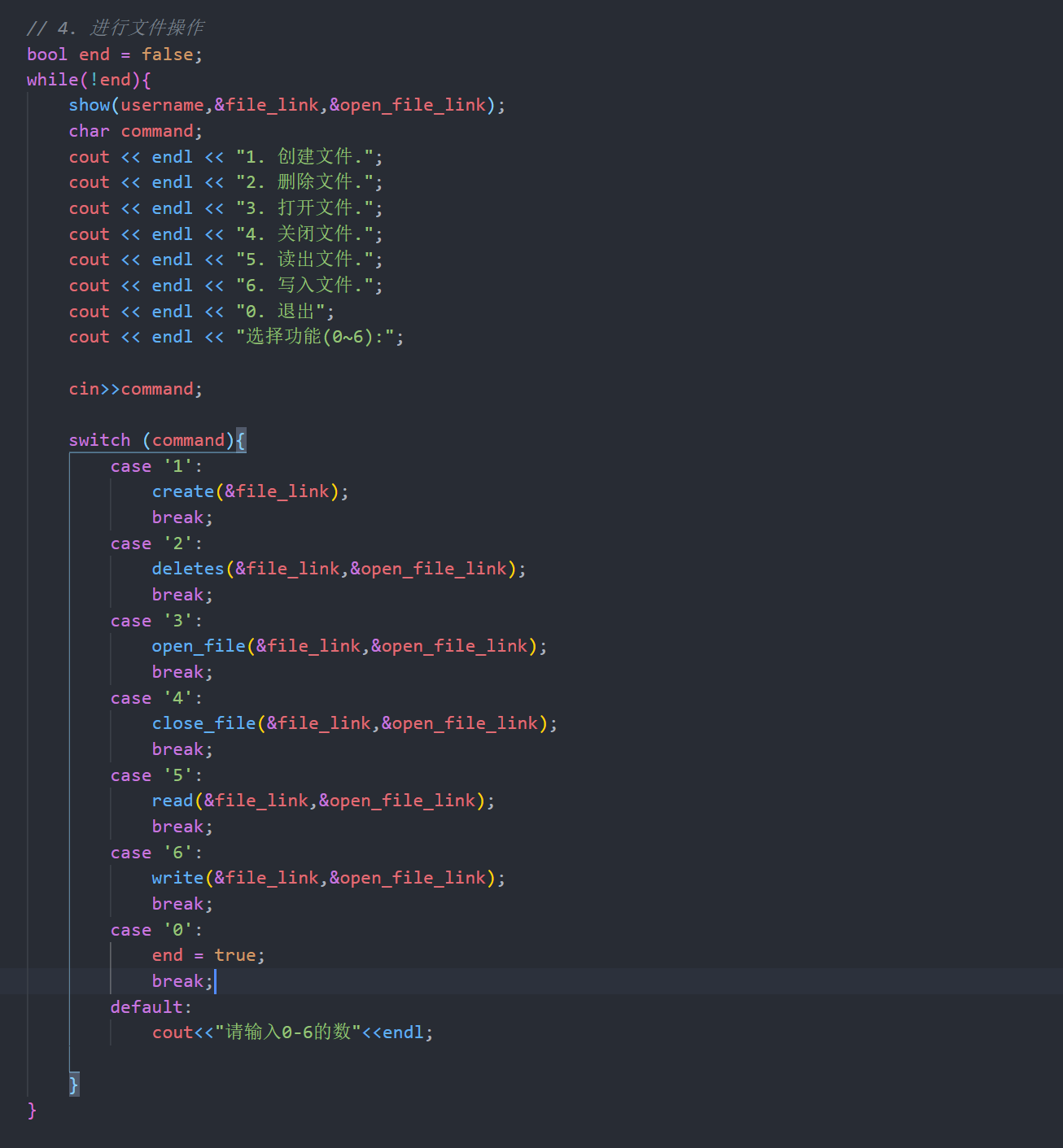




**5. 主程序内容**

**首先，利用Init()函数初始化，并显示菜案。其次，输入用户的姓名。若在用户列表中，则允许用户选择需要的操作。更具用户的输入，调用用户的操作，并显示结果。后附上，每个操作对应的代码，以及main()函数的代码。**





**a. init()：初始化函数。创建user1-user0的用户目录，以及每个用户的10个文件列表**

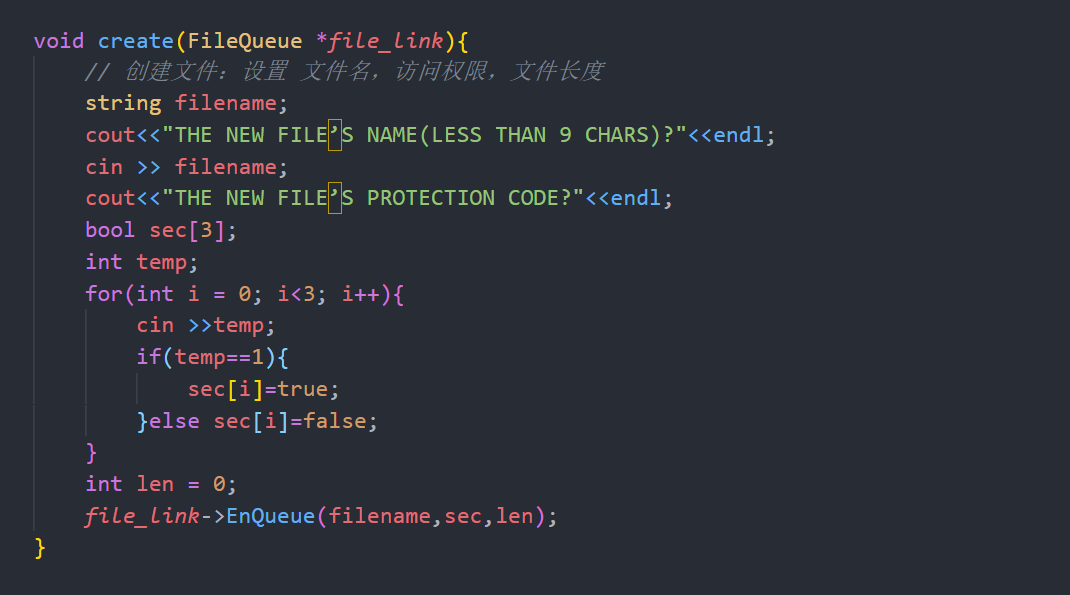


**b. show():显示所有文件和已打开文件的详细信息**

**legal():返回布尔值，判断安全码是否正确，并返回结果。**



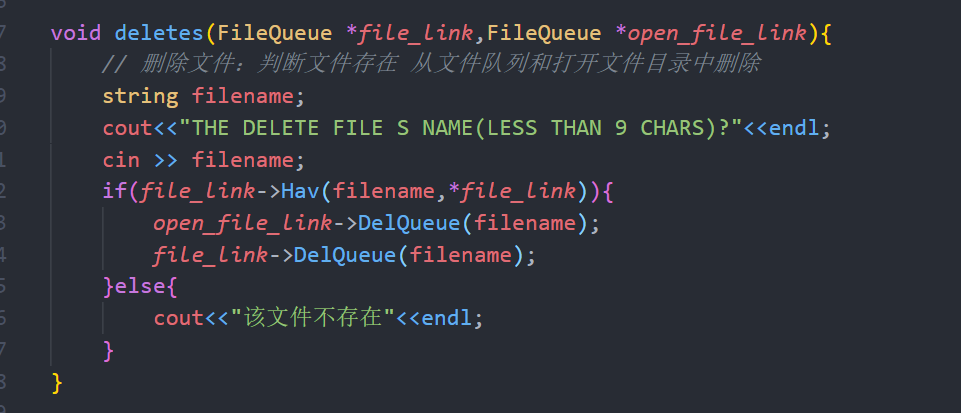
**b. create()方法：创建文件，输入文件名，并且输入安全码。把新建的节点，加入文件目录中。**



c. close\_file():关闭文件，判断文件的状态，满足删除条件，则从打开文件目录中删除改节点。



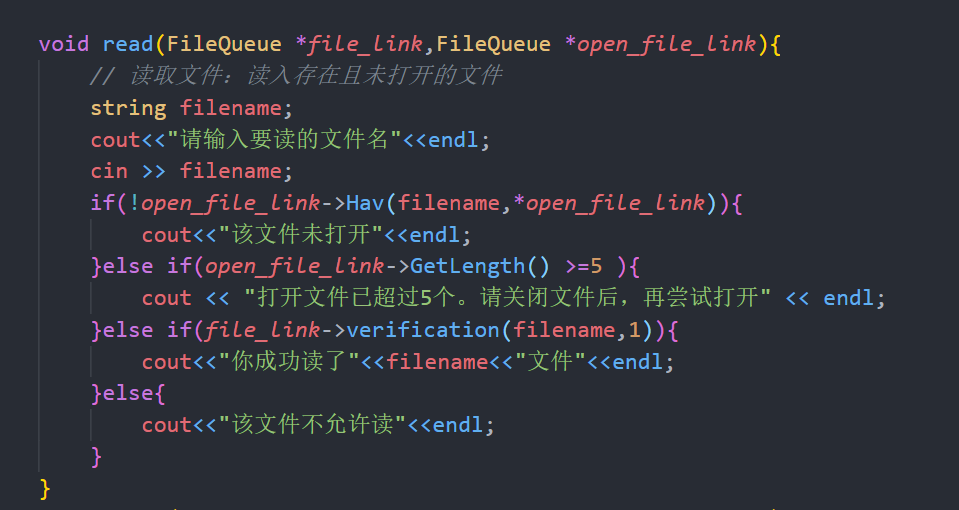
d. 删除文件：判断条件，符合条件，则从文件队列和打开文件目录中都删除该结点。



e. open\_file():打开文件，判断条件，符合，则比较安全码，正确才能打开。并将该节点加入打开文件目录。



f. read():读入文件，判断条件，符合，可以读入文件。

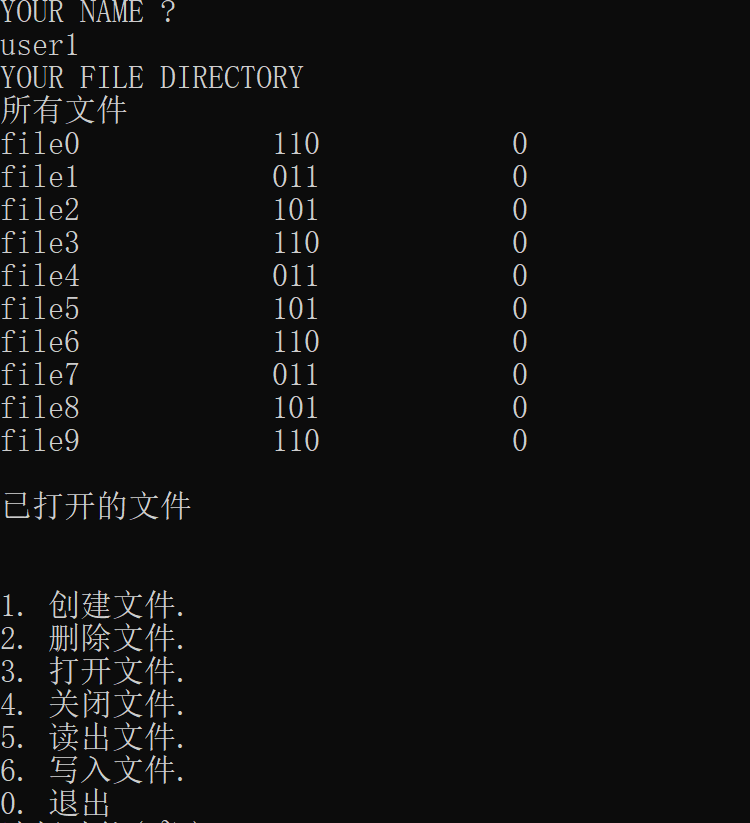


f. write():读入文件，判断条件，符合，可以写入文件。

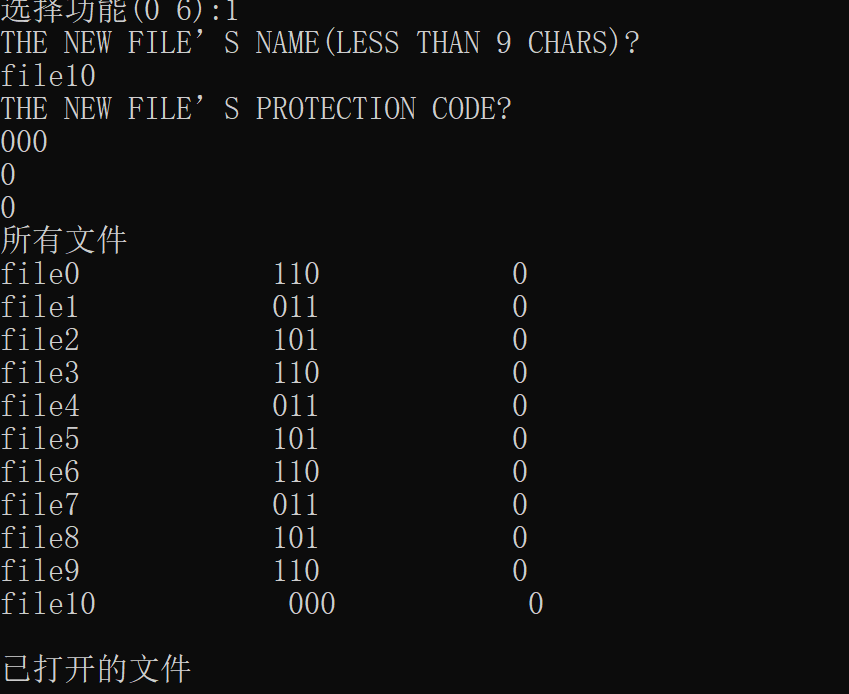


**结果：**

1. 登录系统，显示菜单



2. 创建文档file10,保护码000



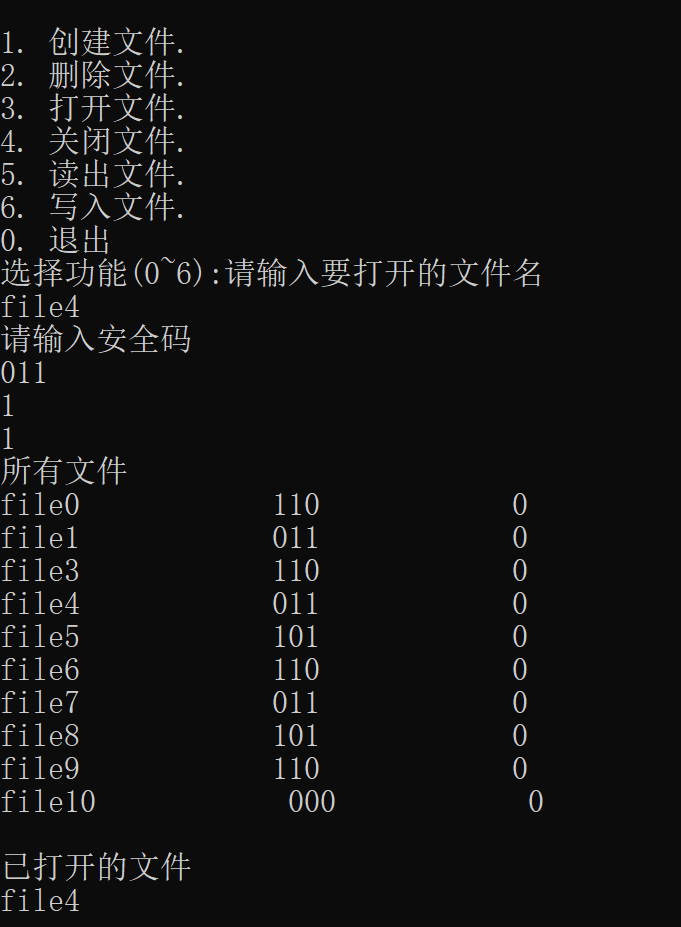
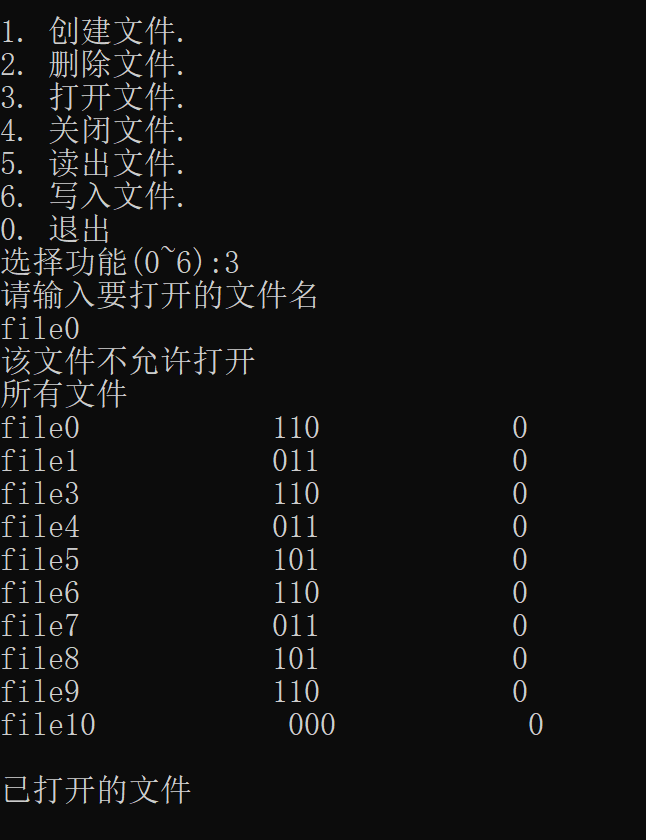
3. 删除文档file2，并显示结果。



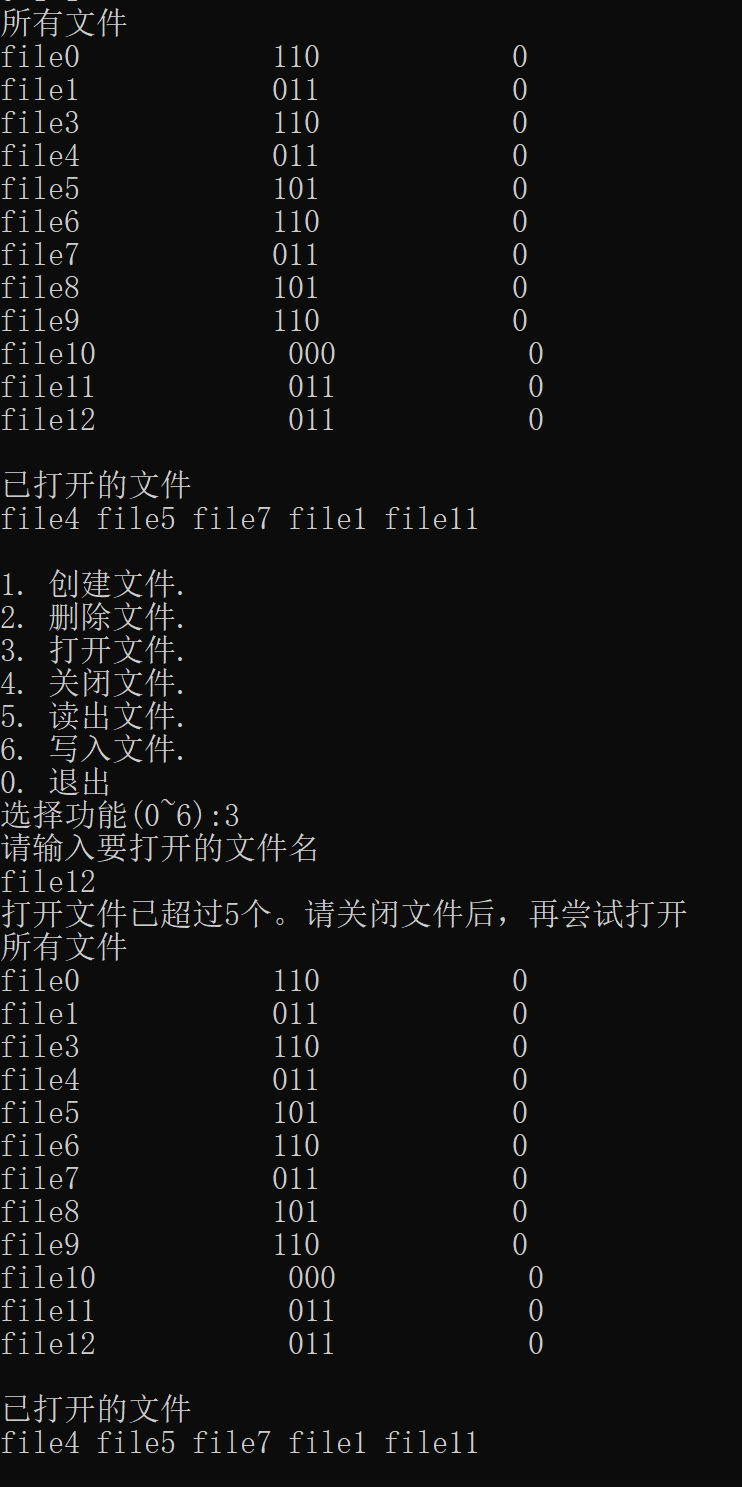
3. 打开文档，需要核对保护码。保护码错误则不能打开，打开文档超过5个则不能打开。不能打开已经打开的文档。

左图：权限不够，无法打开file0

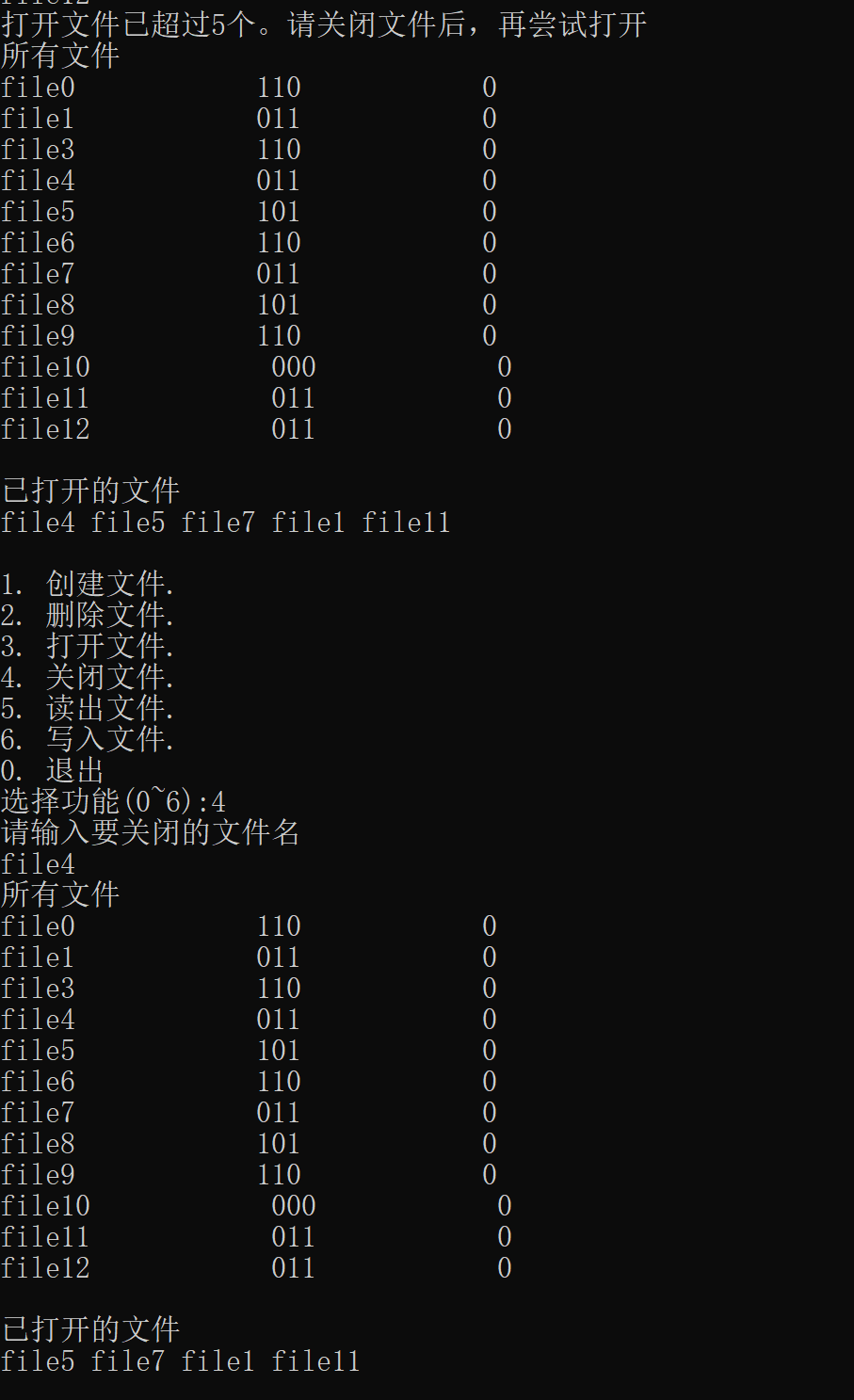
右图：成功打开file4



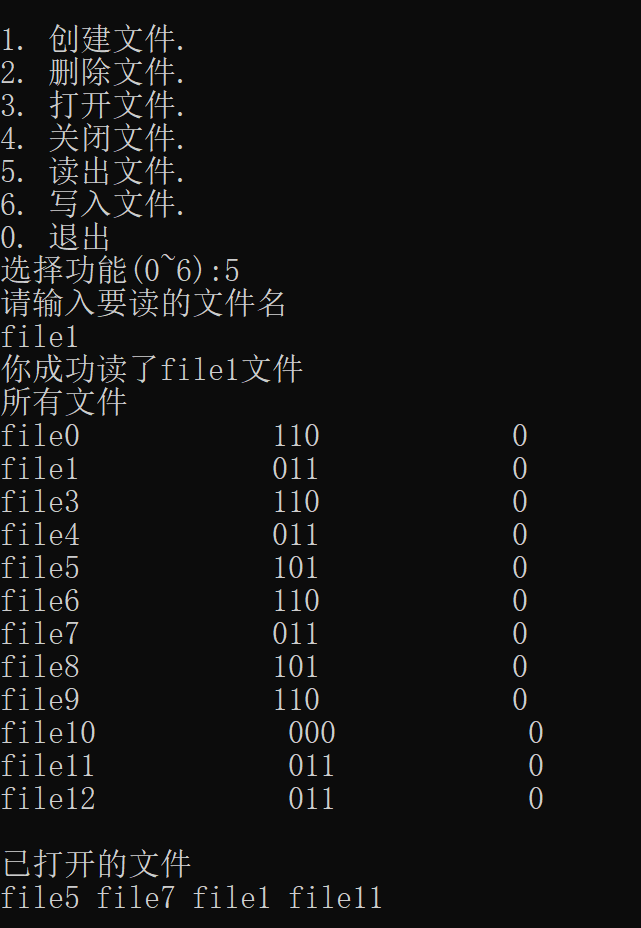
打开文件超过5个，无法继续打开文件。



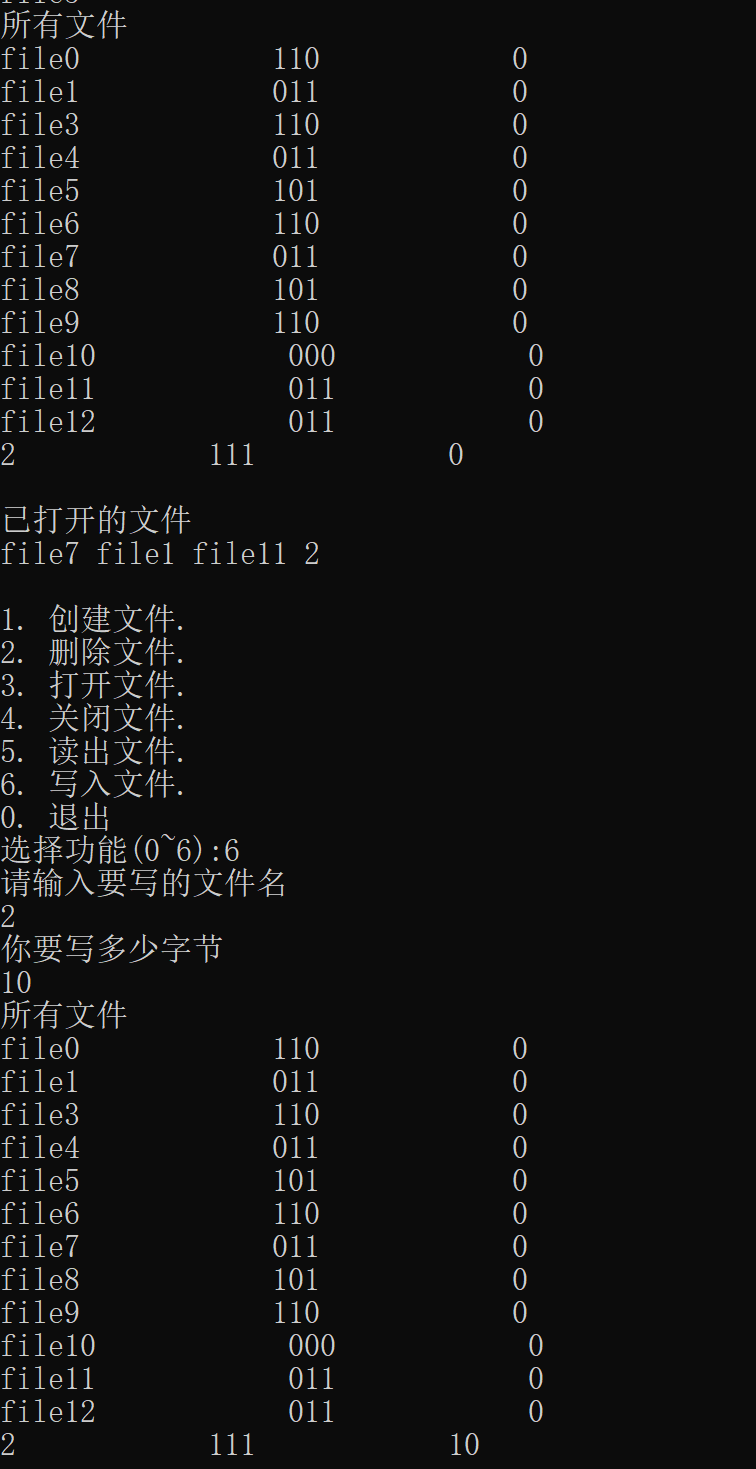
4.关闭文件，file4.



5.读入file1文件。

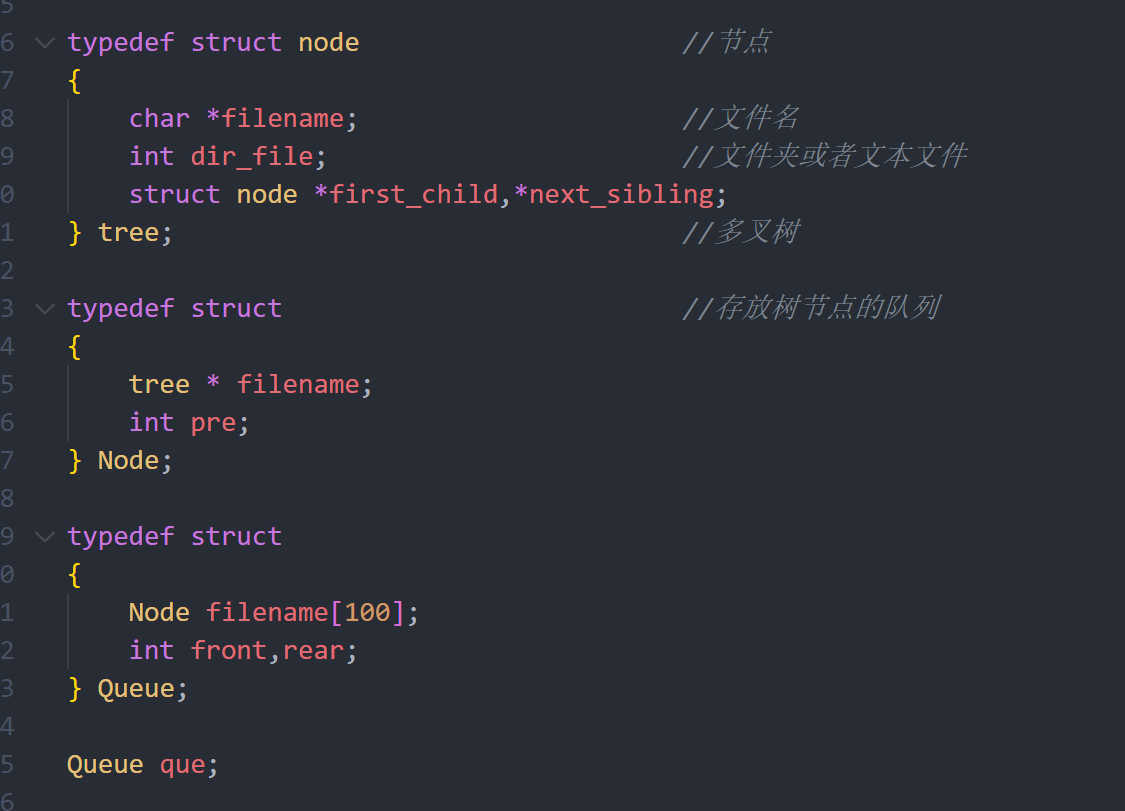


6.写入文件2,写入10个字节。

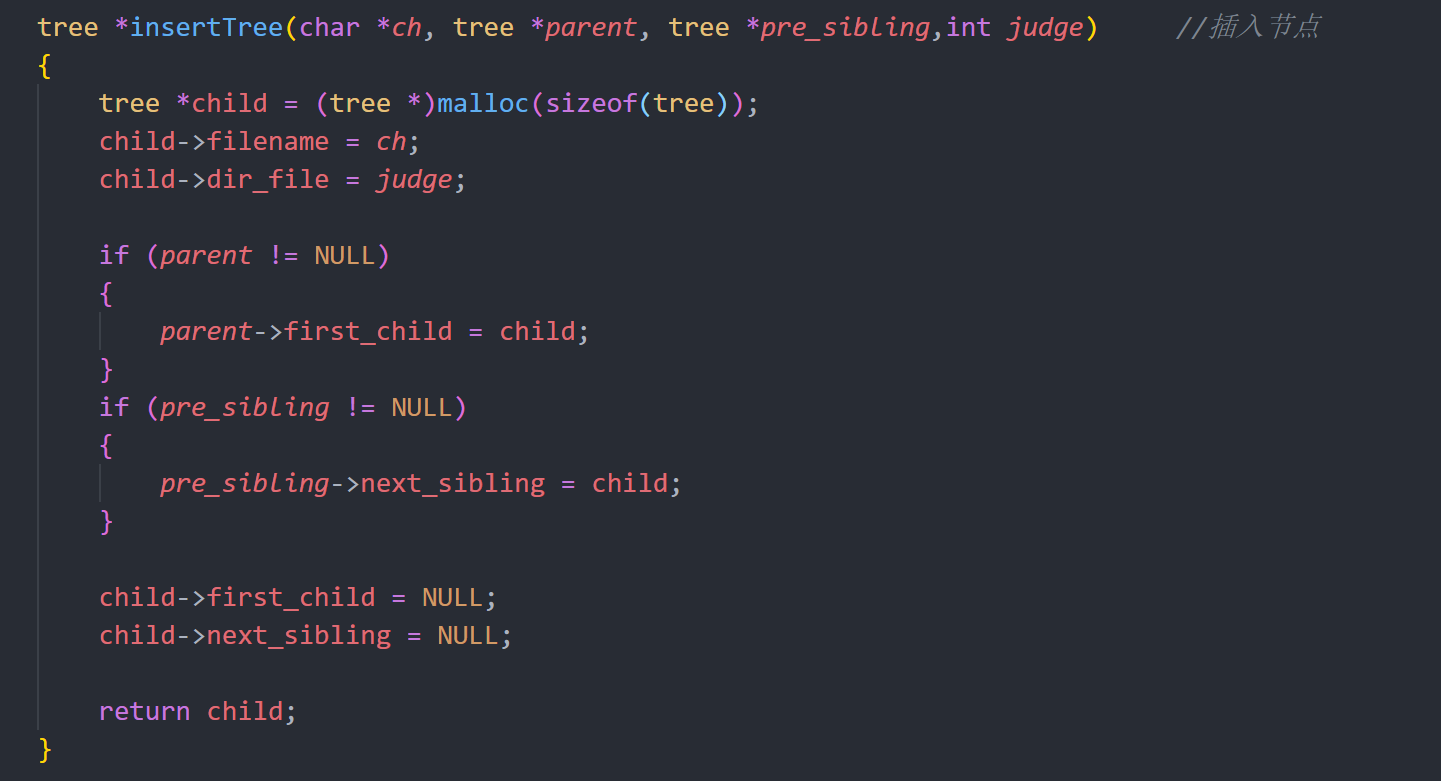


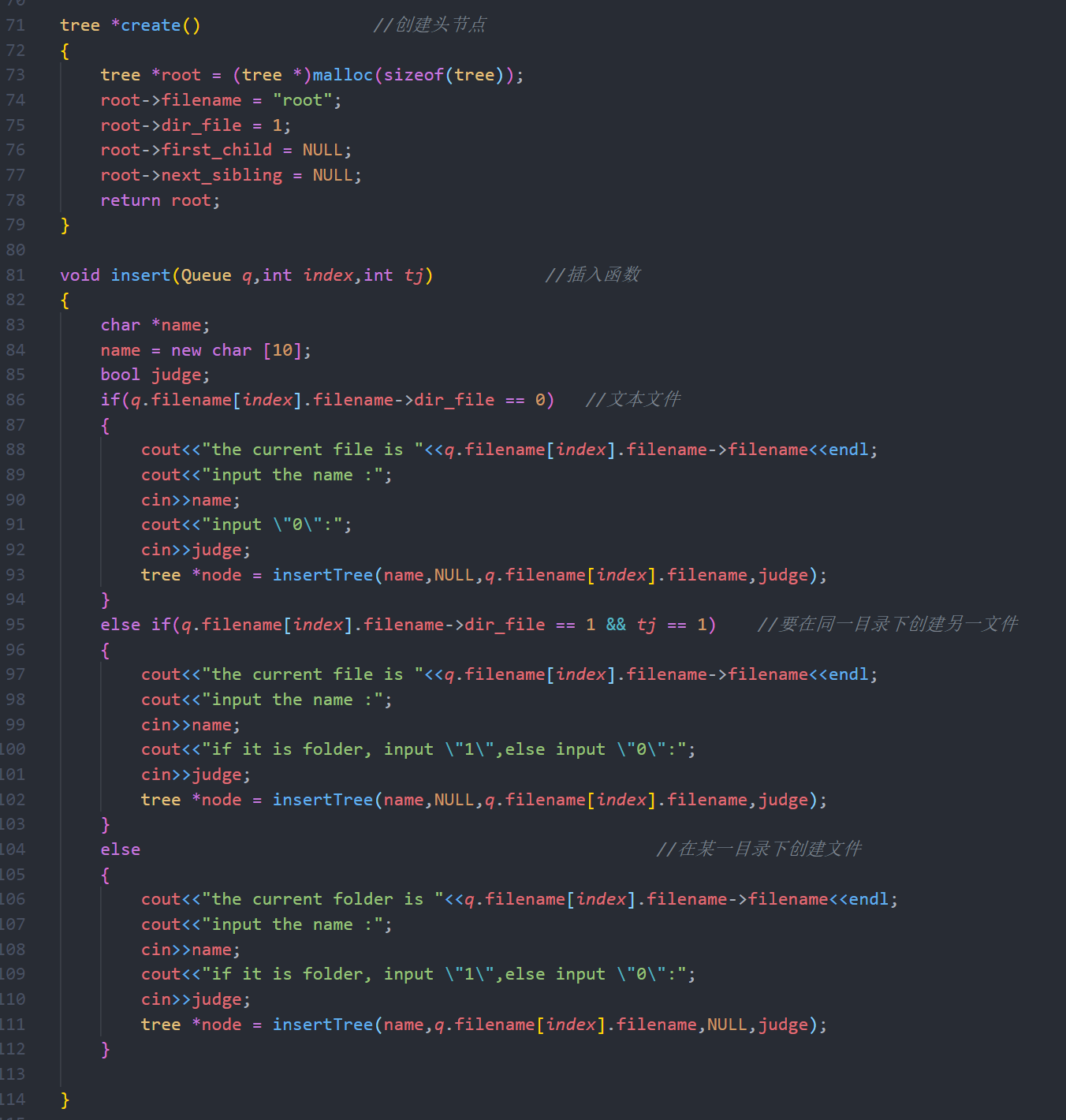
**思考题：**

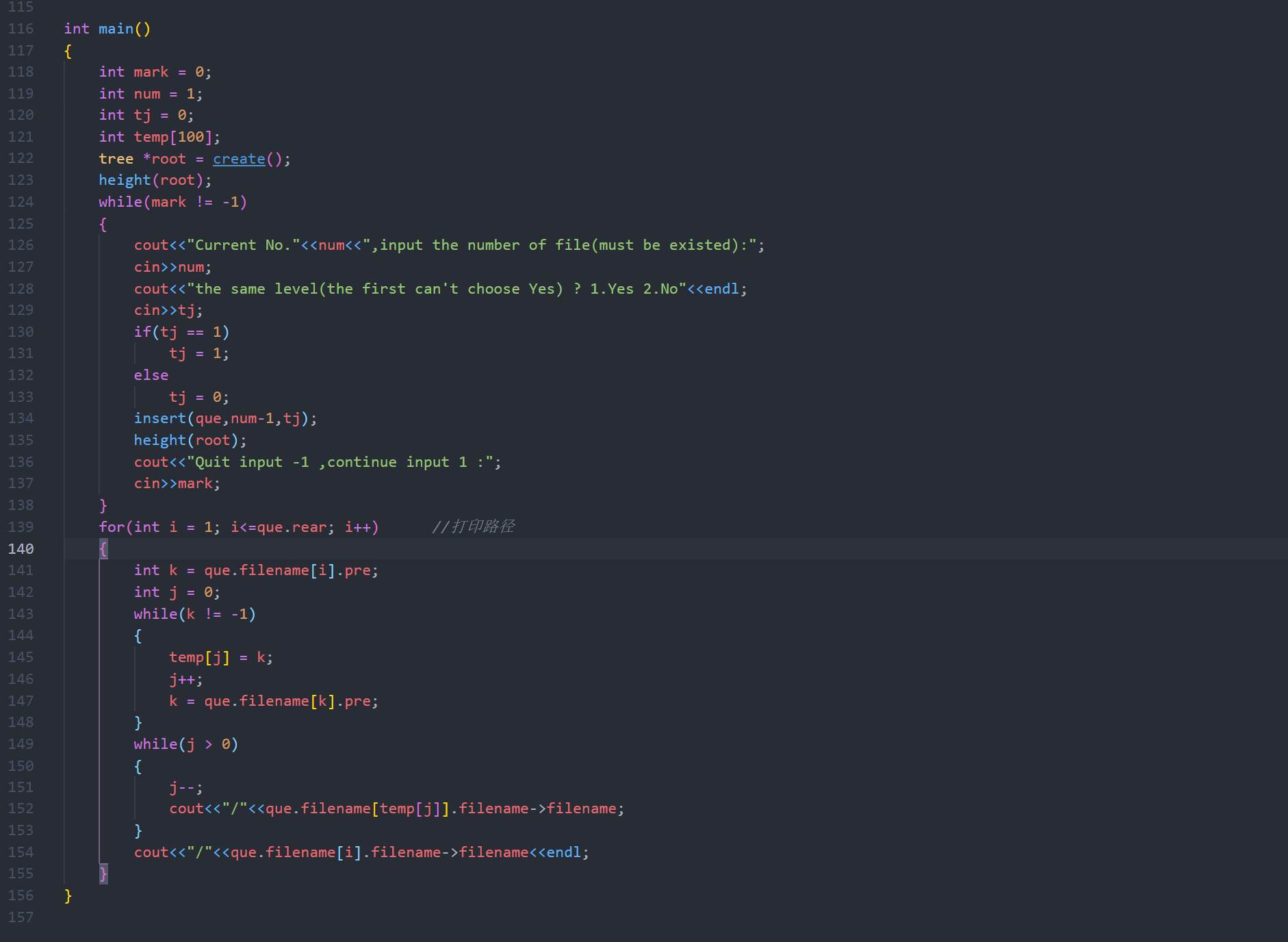
**1. 设计一个树型目录结构的文件系统，其根目录为** root，各分支可以是目录，也可以是文件，最后的叶子都是文件。

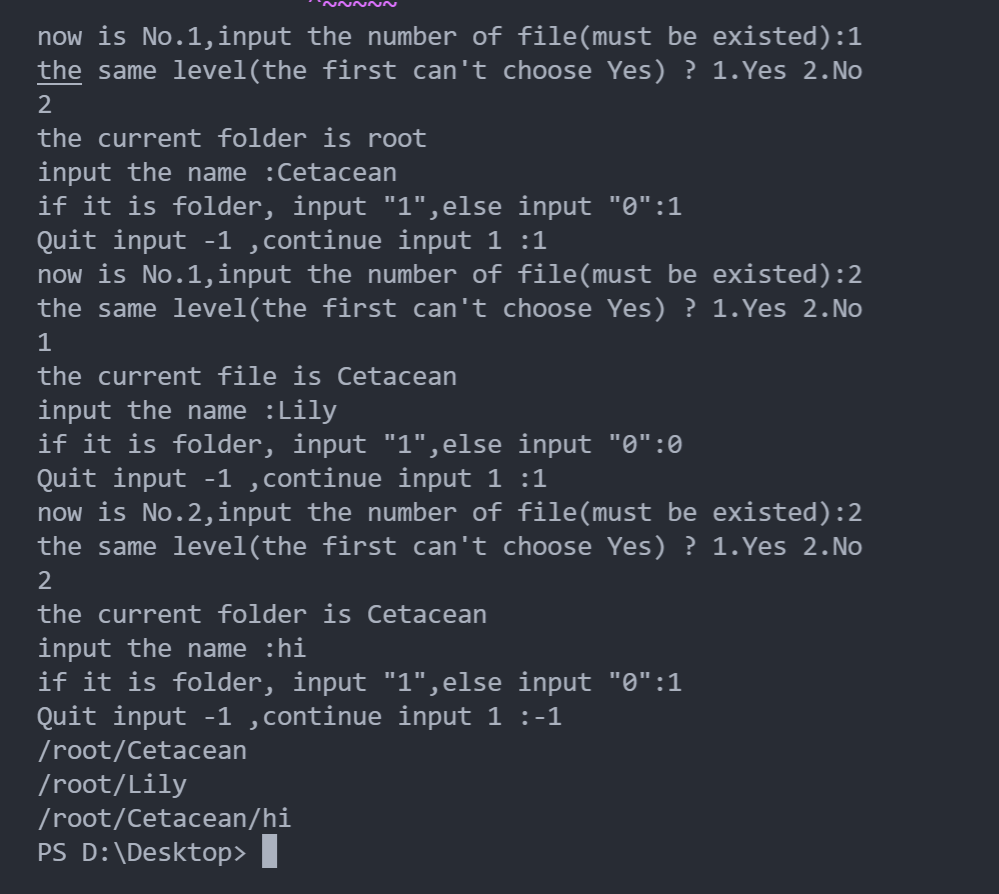












**体会：**

这次实验让我更好的理解了windows和linux中文件管理的原理和过程。同时，也帮助我更好的复习了数据结构课程中学习的有关于链表，结点类的内容。更好的体会到通过选择和合适的数据结构，设计合理的算法流程，就能够编程实现很巧妙的管理系统。触碰到近在咫尺的功能背后的原理，也让我更觉得OS这门课程的趣味性。

**源代码：**

