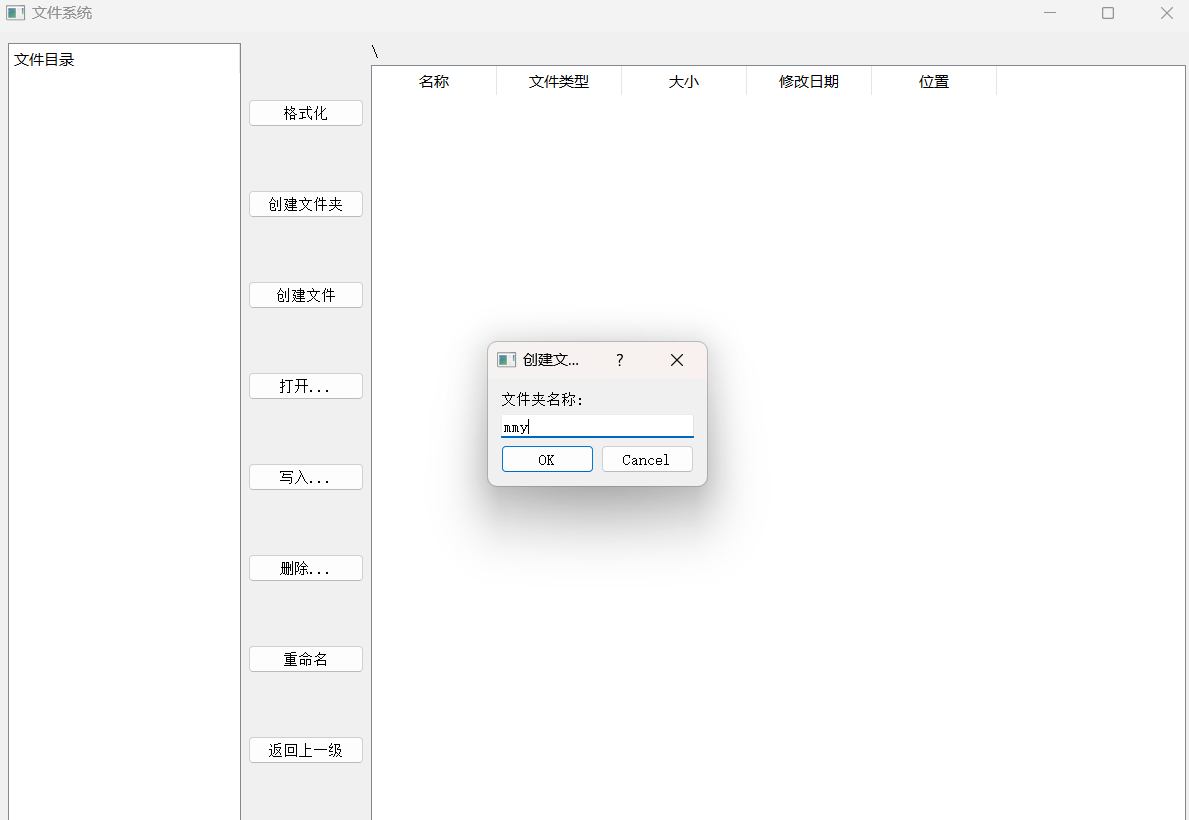
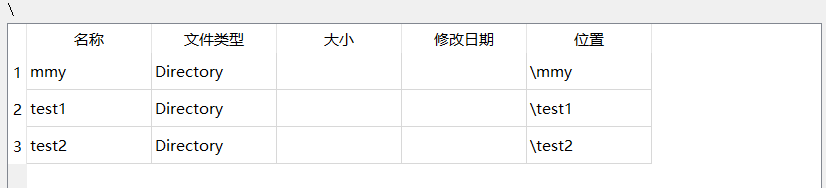
**文件管理系统设计方案报告**

**【开发环境】**

**python3.9+PyQt5**

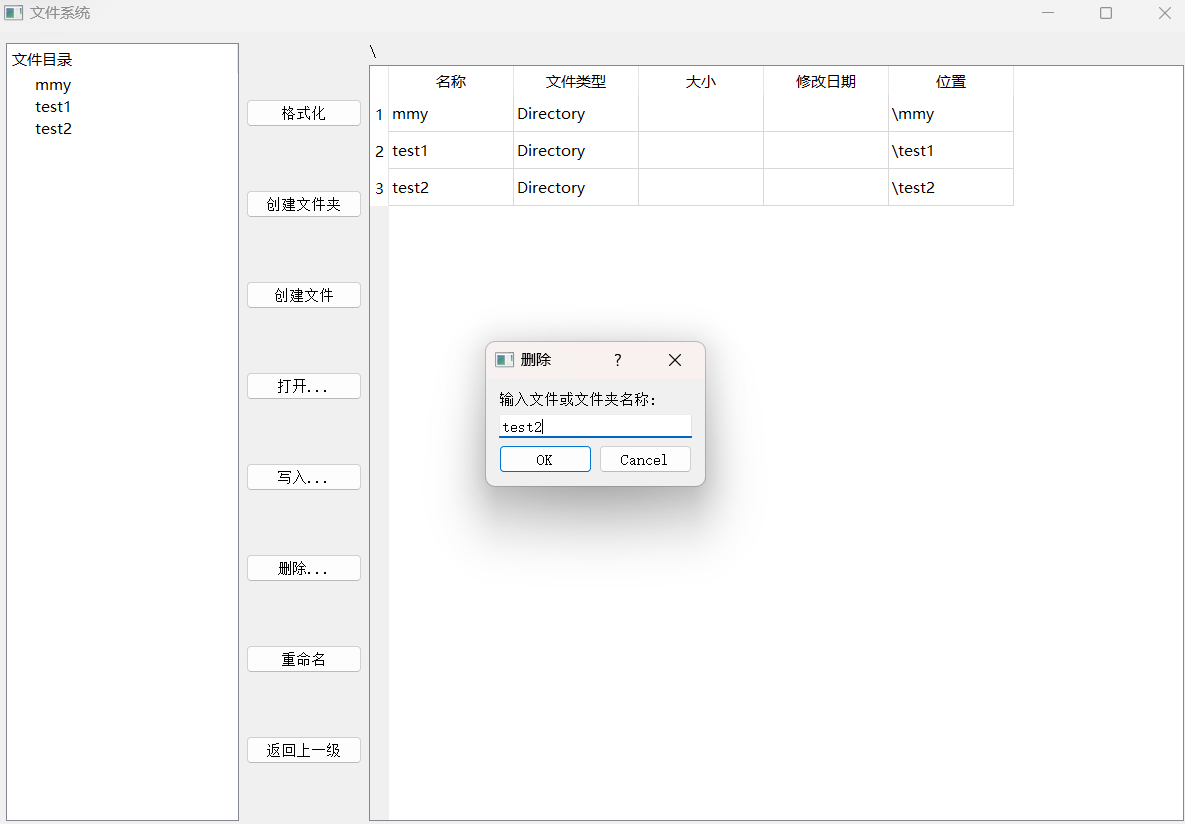
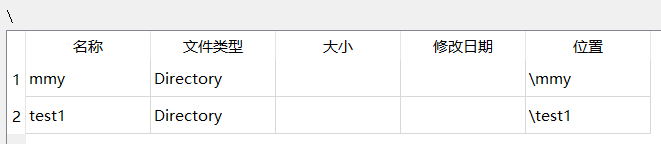
**【功能展示】**

# 创建子目录

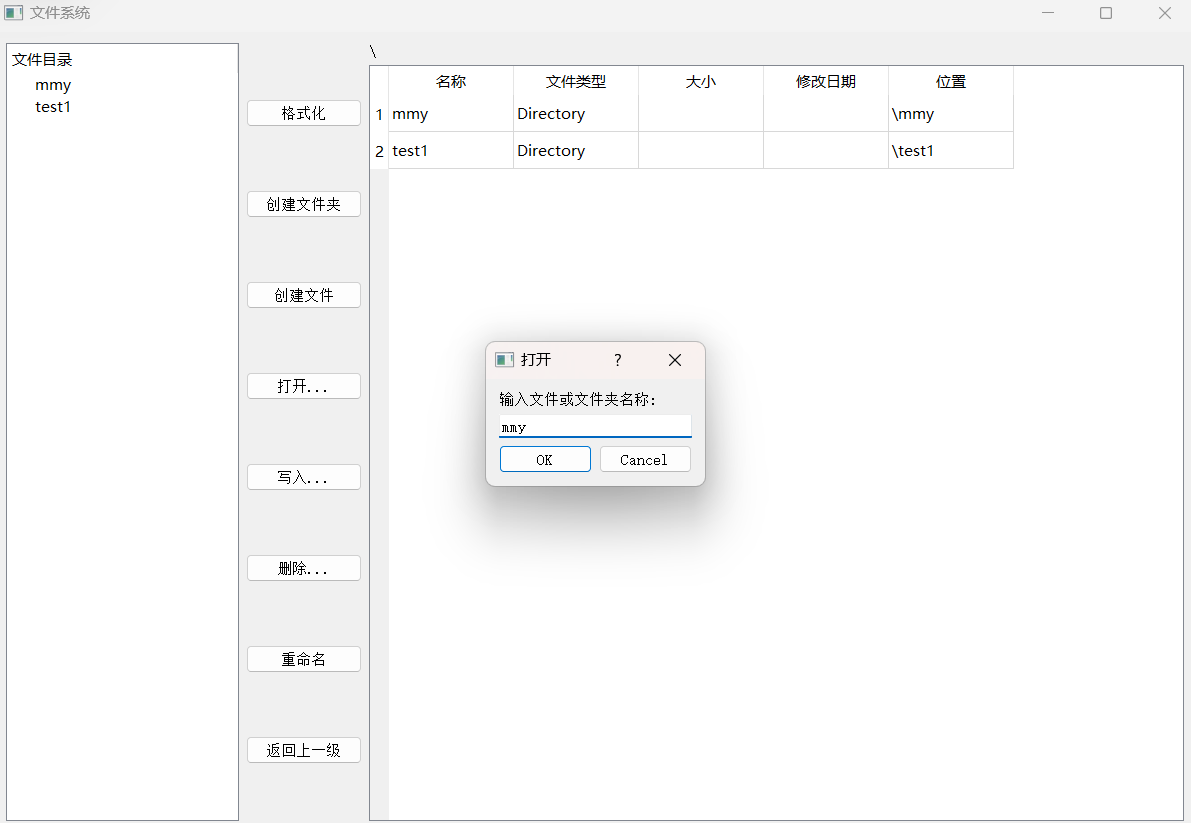
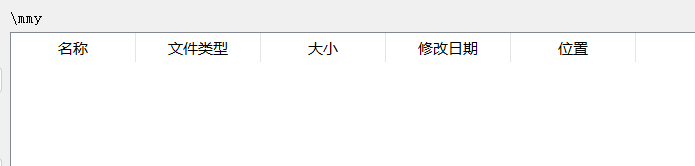
点击”创建文件夹”，以此创建 mmy、 test1、test2文件夹

# 删除子目录

点击”删除…”，输入对应文件夹名称test2，从而删除对应文件夹



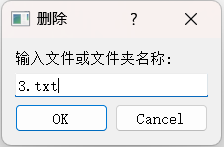
# 更改当前目录

点击“打开…”，输入mmy,进入子目录

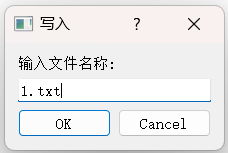
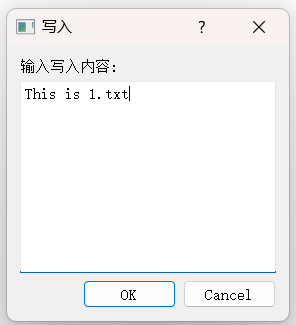
# 创建文件

点击创建文件，相继创建1.txt,2.txt,3.txt.

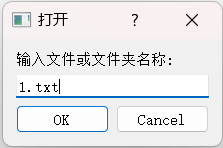
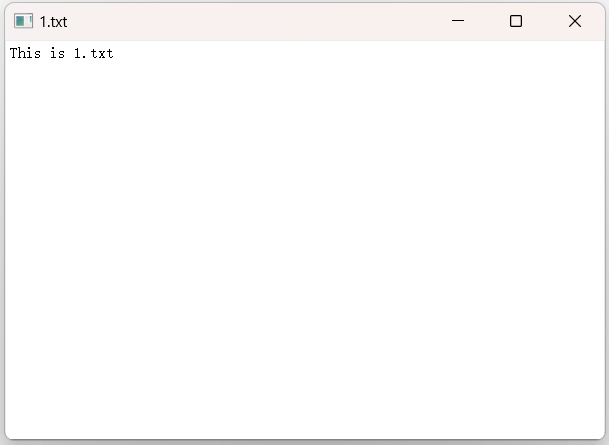
# 删除文件

点击“删除…”，输入需要删除的文件名3.txt

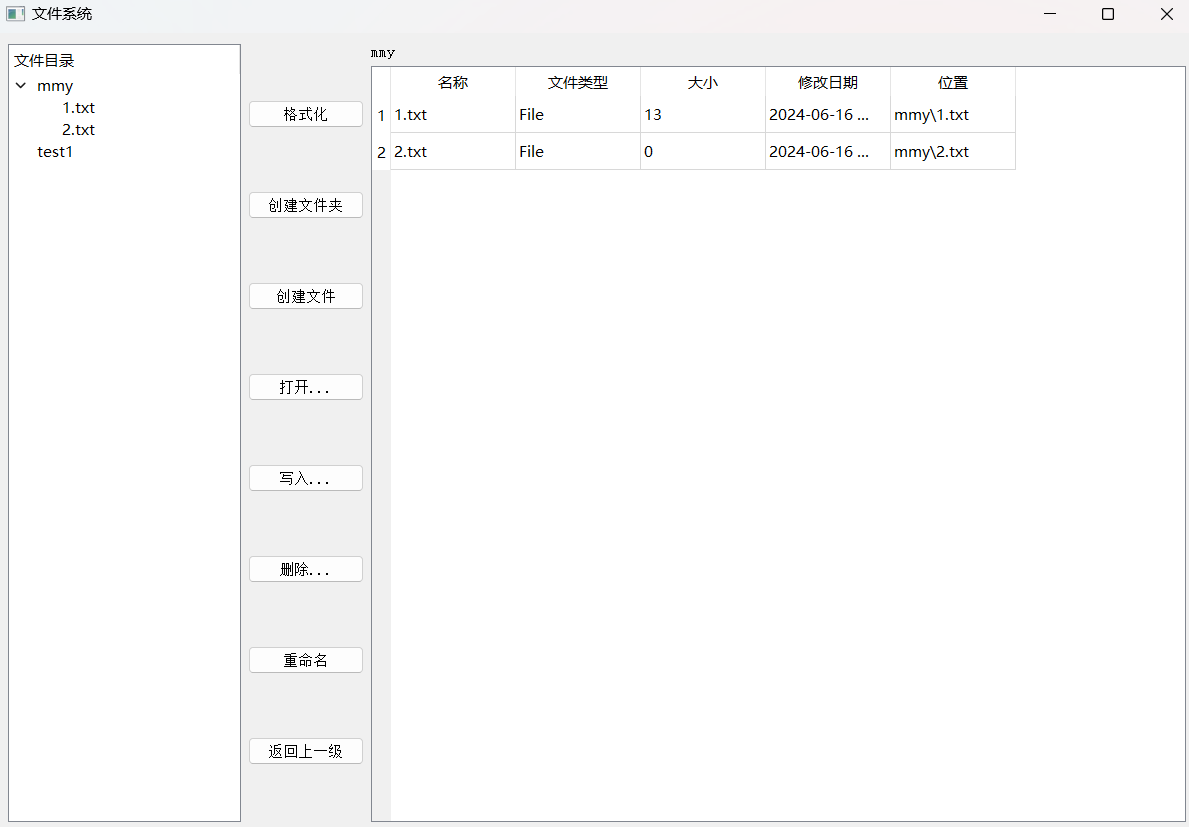
# 写文件

点击“写入…”，依次输入文件名和写入内容

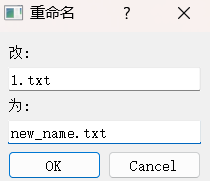
# 读文件

点击“打开…”，输入要打开的文件名

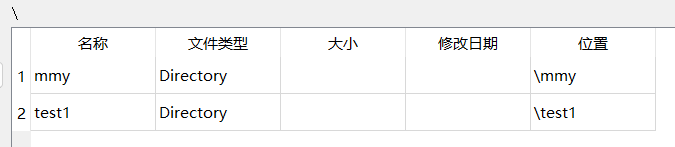
# 显示目录

在界面左侧可以看到目录结构

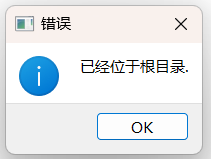
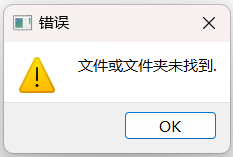
# 重命名

点击重命名，依次输入原来名称和新名称

# 返回上一级目录

点击返回上一级

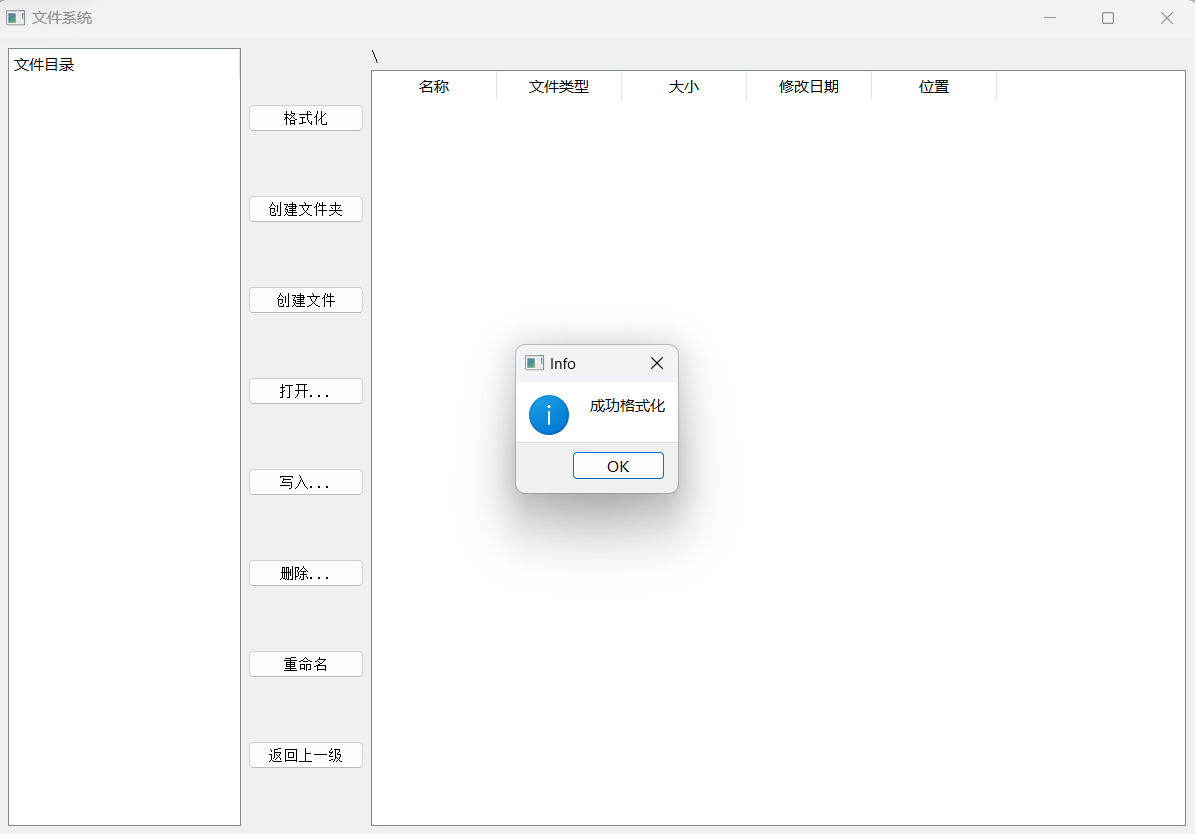
# 错误处理

设置了若干错误提示的窗口

# 保存记录

关闭程序后再次打开，仍能保持上次记录

# 格式化

点击“格式化”，清空磁盘内容

【类的设计】

# DiskBlock

**功能:** 模拟磁盘中的一个块，支持数据的写入、读取和清除。

**主要变量：**block\_size块的大小；data: 存储的数据。

**主要函数:**

write(self, data)

read(self)

clear(self)

**三个函数均针对单个块实现，其中写函数将块大小的数据写入磁盘块，如果超出块大小则截断并返回剩余数据。读函数和清除函数实现较为简单**

# FAT

**功能:** 实现文件分配表，与位图合并，用于管理文件在磁盘块上的存储位置以及每个磁盘块的状态。具体实现为：①当某项为-2时，表示对应磁盘块为空；②当某项为-1时，表示一个文件存储结束的标志；③当为其他数字时，表示文件存储的下一个磁盘块。

**主要变量：**table 一个列表，每个元素表示对应块的状态或指向下一个块的索引

**主要函数：**

find\_blank(self)：顺序扫描table表，寻找一个空的磁盘块

write(self, data, disk)：写入文件。函数内会维护文件开始写入磁盘块以及当前写入的磁盘块的索引，利用DiskBlock类中对单个磁盘块的写入函数实现对整个文件的写入，同时维护FAT表

delete(self, start, disk)：删除文件。首先找到文件的起始磁盘号，根据FAT表以及DiskBlock类中的对单个磁盘块的清空逐渐完成对整个文件的删除，同时置FAT表对应位置为-2

read(self, start, disk)：读文件。首先找到文件的起始磁盘号，根据FAT表以及DiskBlock类中的对单个磁盘块的读逐渐完成对整个文件的读取，

# FCB

**功能:（**File Control Block, FCB,文件控制块）类用于管理文件的元数据。每个文件或目录在文件系统中都有一个对应的FCB，记录文件或目录的基本信息，本项目中记录了文件的名称、类型（文件或目录）、大小、最后修改时间以及路径等。

**主要变量：**

name: 文件或目录的名称。

is\_directory: 布尔值，表示是文件（False）还是目录（True）。

first\_block: 文件的起始块索引。

size: 文件大小（以字节为单位）。

modification\_time: 文件或目录的最后修改时间

# FileSYStem

**功能:** 文件系统类（FileSystem）负责管理整个文件系统的操作，包括文件和目录的创建、删除、重命名、读取和写入等操作。该类利用文件分配表（FAT）和磁盘块（DiskBlock）来实现这些功能。

**主要变量：**

disk\_name: 磁盘文件名。

block\_size: 磁盘块的大小。

num\_blocks: 磁盘块的数量。

fat: 文件分配表对象。

disk: 磁盘块列表。

current\_dir: 当前目录。

directory: 目录结构。

**主要函数：**

create\_subdirectory(self, name)：在当前目录下创建子目录。首先创建FCB对象，设置为目录类型，然后在当前目录下添加新目录项

delete\_subdirectory(self, name)：删除当前目录下的子目录。首先在当前目录下查找并删除目录项，然后从目录结构中移除对应子目录。

create\_file(self, name, data)：在当前目录下创建文件并写入数据。要将文件数据到磁盘，获取起始块索引。然后创建文件的FCB对象，并在当前目录下添加新文件项。

write\_file(self, name, data) ：写入文件数据。首先在当前目录下查找文件的FCB对象，然后删除旧数据，写入新数据，更新FCB对象。

read\_file(self, name)：读取文件数据。首先在当前目录下查找文件的FCB对象，然后就可以调用fat的读函数进行读取

delete\_file(self, name)：删除当前目录下名为name的文件。首先在当前目录下查找文件的FCB对象，然后删除文件数据，从FAT表中移除文件的块，最后要从从目录中移除文件项。

# FileSYStemAPP

**功能：在该类中实现了GUI界面的设计，设计细节在此不再赘述。**