# 文件管理系统项目文档

**2151601 邹昕恺**

1. **项目概述**
2. 基本要求：在内存中开辟一个空间作为文件存储器，在其上实现一个简单的文件系统，退出这个文件系统时，需要该文件系统的内容保存到磁盘上，以便下次可以将其回复到内存中来。
3. 具体要求：文件存储空间管理可采取链接结构（如FAT文件系统中的显式链接等）或者其他学过的方法；空闲空间管理可采用位图或者其他方法；文件目录采用多级目录结构，目录项目中应包含：文件名、物理地址、长度等信息。
4. 本程序通过索引方式管理存储空间，并采用位图的形式管理空闲空间；同时，本程序使用多级目录结构管理文件目录、FCB记录文件名、物理索引首地址、文件长度。
5. 程序在运行的途中会在C盘目录下创建2151601文件夹（由于每台电脑的磁盘分配不尽相同，但每台电脑都存在C盘），同时在这个目录下同步记录模拟文件系统的内容，重新运行程序时，之前文件系统的内容会保留在内存中。
6. 本程序只是作为操作系统中的一种文件管理方式，根目录为C:\2151601，并且作为用户无法访问根目录。仅能从程序将磁盘分成的X盘、Y盘、Z盘中进行操作，并且用户不能增加盘以及删除盘。
7. 本程序内存所分配的磁盘数组空间很小，这样可以更方便用户理解当磁盘空间不足时，磁盘的运作模式。
8. **特别说明**

请按照下图的方式进行控制框的各个属性值，尤其是旧版控制框以及取消快速编辑模式：

1. **项目概述**

Windows 10

Visual Studio 2022

C++

1. **程序算法实现**
2. **多级目录**

二叉树结构可以被用来实现多级目录，其中每个节点代表一个目录或文件。下面我将用文字说明如何通过二叉树结构实现多级目录。

根节点：根节点代表整个目录结构的根目录。它是唯一的起始点，没有父节点。

目录节点：每个目录节点代表一个目录。它包含一个名称和一个指向左子节点和右子节点的指针。与许多通过二叉树数据结构实现多级目录的区别在于，该二叉树将同级目录链接于右子树，子目录和文件链接于左子树。

通过这样的结构，我们可以建立一个层级结构，实现多级目录的存储和访问。

**目录结构**如下图所示：

|2151601

----|X

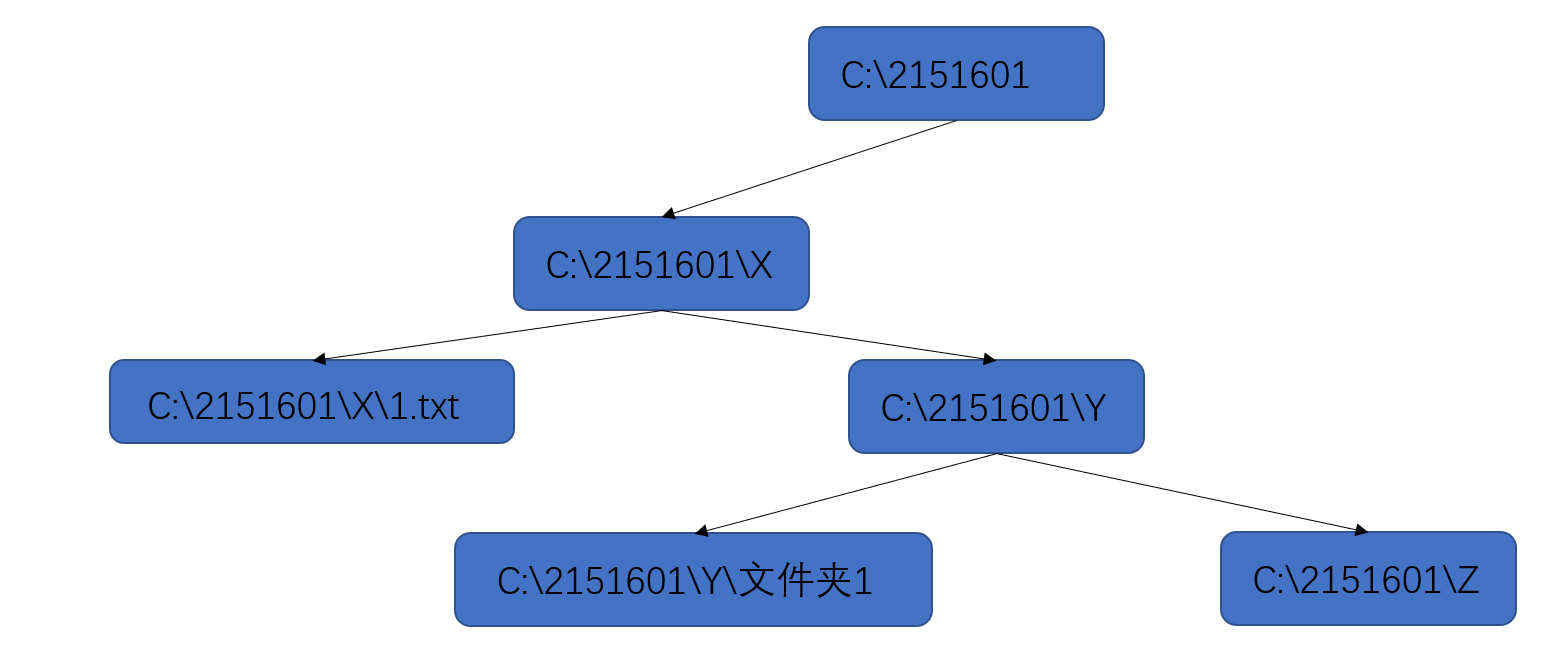
----|----|1.txt

----|----|2

----|----|----|1.txt

----|Y

----|Z

**二叉树数据结构图**如下：

FCB文件控制块记录了文件的名称、物理索引首地址和文件长度等信息。另外，从程序实现将模拟文件系统的内容同步到计算机磁盘，并在下次执行程序时恢复上次的内容到内存中的功能角度来看，基于二叉树数据结构，程序需要额外记录每个文件或文件夹在计算机磁盘中的绝对路径、该文件或文件夹的父节点在计算机磁盘中的绝对路径，以及该文件或文件夹与父节点的关系（是同级关系，链接于右子树，记录为1，还是父子关系，链接于左子树，记录为0）。

在将模拟文件系统的内容存储到计算机磁盘时，可以按照前序遍历二叉树的顺序写文件。例如，在计算机磁盘中，文件路径为 "C:\2151601\目录文件.txt"。这样做可以确保在下次运行模拟程序并恢复上次的文件系统目录结构时，按照从前到后读取文件的顺序，当需要插入一个子节点时，其父节点已经被建立。

通过这样的记录方式和存储顺序，可以有效地保证文件系统的一致性和完整性，以及在恢复文件系统时的正确性。

目录文件中对某一个文件记录的信息从左到右依次为：文件名、物理块中的索引首地址、文件长度、文件在计算机的绝对路径、该文件的上级目录。

1. **用索引方式管理存储空间**

索引方式是一种常见的管理存储空间的方法，它使用索引数据结构来维护文件和其存储位置之间的映射关系。索引节点是一个特殊的数据结构，用于管理文件和其存储位置之间的映射关系。每个文件对应一个索引节点，它包含了文件的元数据信息和存储位置的指针。主索引表是一个高效的数据结构，用于存储所有文件的索引节点的位置信息。主索引表通常存储在内存中，以便快速访问。为实现存储空间的管理，可以将存储空间划分为固定大小的块（block）。索引节点中的指针指向文件在存储空间中占用的块的起始位置。当创建一个新文件时，分配足够的块给该文件，并创建相应的索引节点。修改文件时，根据需要增加或删除块，并更新索引节点的指针。同时通过主索引表可以快速找到指定文件的索引节点。索引节点中的指针提供了访问文件存储位置的途径。使用索引方式管理存储空间的好处是能够快速定位文件的存储位置，减少了文件的查找时间。此外，索引方式还可以支持文件的动态修改和扩展，使得文件管理更加灵活和高效。

为了实现存储空间的管理，程序会在内存中创建一个大小为100×4的二维数组。这个数组的含义如下：

1. 数组中的每个元素代表一个物理存储块。
2. 一共有100个物理存储块，每个存储块占用数组中的一行。
3. 在程序模拟过程中，该数组不直接记录文件的实际内容，而是使用特定的数字来表示不同的情况。
4. 数字(-2)表示该物理块中存储的内容是文件内容。这意味着当需要查看文件内容时，程序会根据目录文件中记录的绝对路径信息来访问计算机磁盘中的文件，并将文件内容读入内存。
5. 数字(-1)表示该物理块为空，没有记录任何信息。
6. 数字0-99表示索引值，用于索引其他的物理存储块。

这样，二维数组中的内容会与计算机磁盘中的文件"C:\2151601\磁盘文件.txt"保持同步。下次运行程序时，程序将根据磁盘文件中的内容来恢复二维数组的状态。

1. **用位图方式管理空闲空间**

位图方式是一种常见的管理空闲空间的方法，它使用位图数据结构来跟踪存储空间中每个块的使用情况。以下是我用位图方式管理空闲空间的大致步骤，主要分为存储空间划分，位图数据结构，空闲空间管理，分配空闲块，释放块，查找空闲块：

1. 存储空间划分：将整个存储空间划分为固定大小的块，每个块表示一个存储单元。可以将每个块编号，从0到N-1，其中N是存储空间中块的总数。
2. 位图数据结构：创建一个位图，其大小与存储空间中块的总数相匹配。每个位（bit）对应一个块，用于表示该块的使用状态。例如，0表示空闲，1表示已使用。
3. 空闲空间管理：初始状态下，所有位都设置为0，表示所有块都是空闲的。
4. 分配空闲块：当需要分配一个块时，从位图中找到第一个值为0的位，并将其设置为1，表示该块已被使用。同时，记录该块的编号作为分配的空闲块。
5. 释放块：当不再需要某个块时，将对应位图中的位设置为0，表示该块变为空闲。
6. 空闲块的查找：当需要空闲块时，遍历位图，找到值为0的位所对应的块编号，即为一个空闲块。

使用位图方式管理空闲空间的好处是可以快速查找空闲块，避免了线性搜索的开销。此外，位图还可以用于快速统计空闲块数量和连续空闲块的情况。使用位图方式管理空闲空间的好处是能够快速查找空闲块和占用块的状态，以及有效地利用存储空间。位图文件的存储可以保证数据的持久化，使得下次运行程序时可以正确地恢复位图数组的状态。

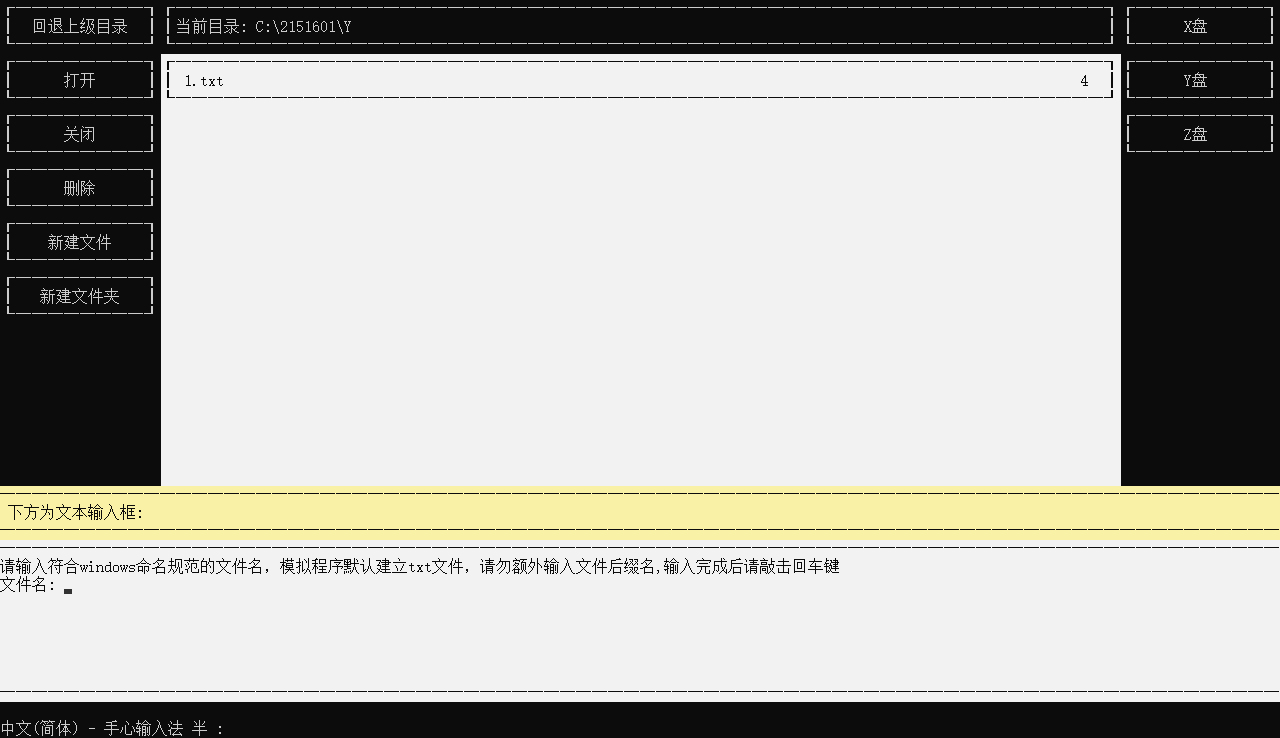
为了管理空闲空间，程序采用了一个大小为100的一维数组来实现位图。该数组的含义如下：

1. 数组中的每个元素代表一个物理存储块。
2. 数组的初始状态为0，表示所有物理块都是空闲的。
3. 当某个物理块被占用时，相应的数组元素会被设置为1，表示该物理块已被占用。

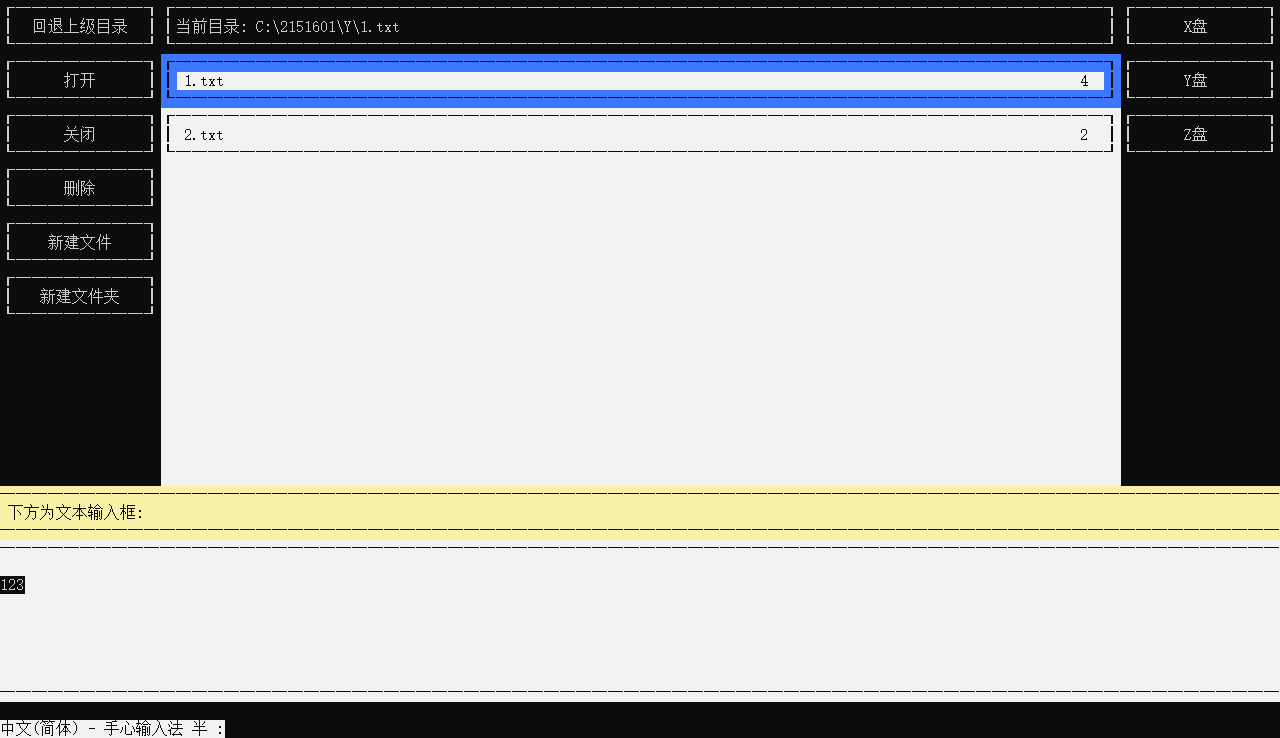
同时，位图数组中的内容会与计算机磁盘中的位图文件"C:\2151601\位图文件.txt"保持同步。下次运行程序时，程序将根据位图文件中的内容来恢复位图数组的状态。

1. **运行注意事项：**
2. 您在建立文件或文件夹时的命名必须符合Windows命名规范，且同一目录下不可以出现同名文件或文件夹，否则本程序会提示命名不合法，直到您输入符合规范的命名。
3. 本程序默认建立txt文件，如果你想要建立一个txt文件，则不需要输入文件后缀，输入文件后缀反而会被程序判定为命名不合法。
4. 本程序只要求同一目录下文件名不重复即可，不同目录下可以有重名文件，此外，本程序不认为文件夹和文件重名，例如：文件1.txt和文件夹1可以在同一目录下存在。
5. **模拟界面说明：**
6. **进入程序界面：**

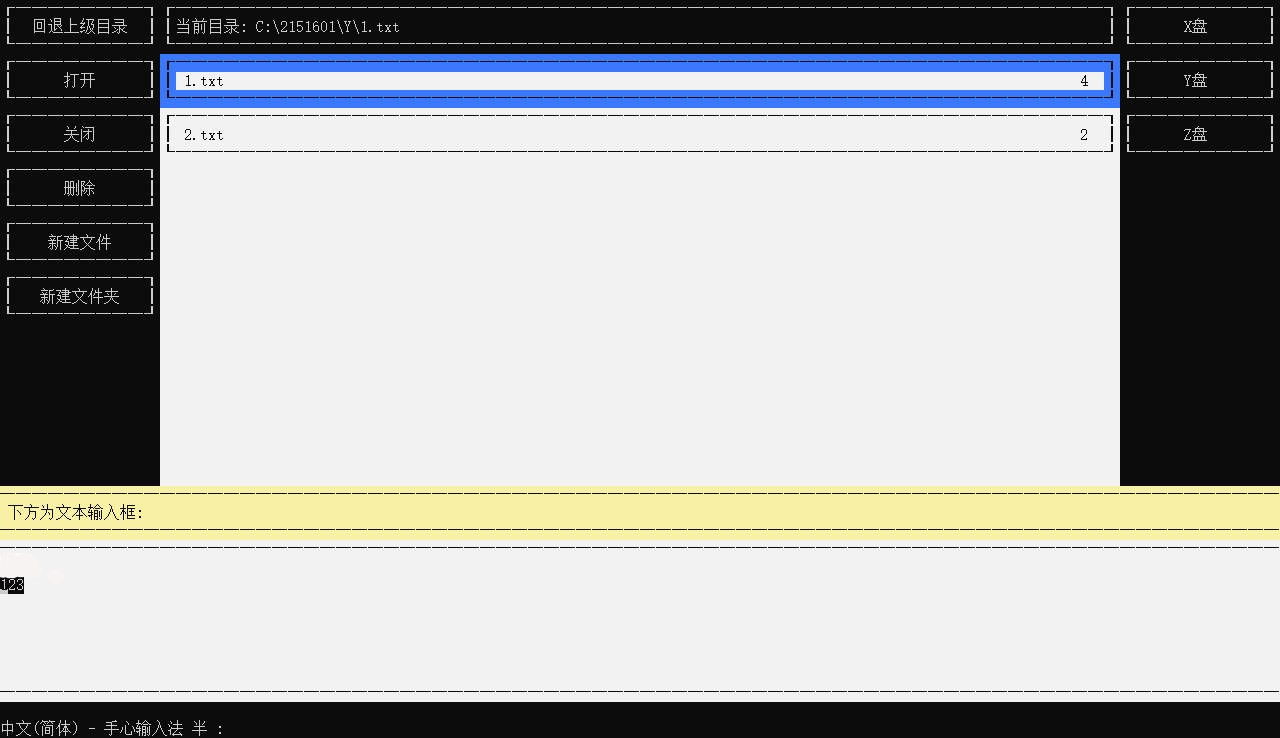
菜单的右边可以选择根目录底下的X盘、Y盘、Z盘，菜单的左边为对目录以及文件的操作。

1. **选择一个盘，新建一个文件**

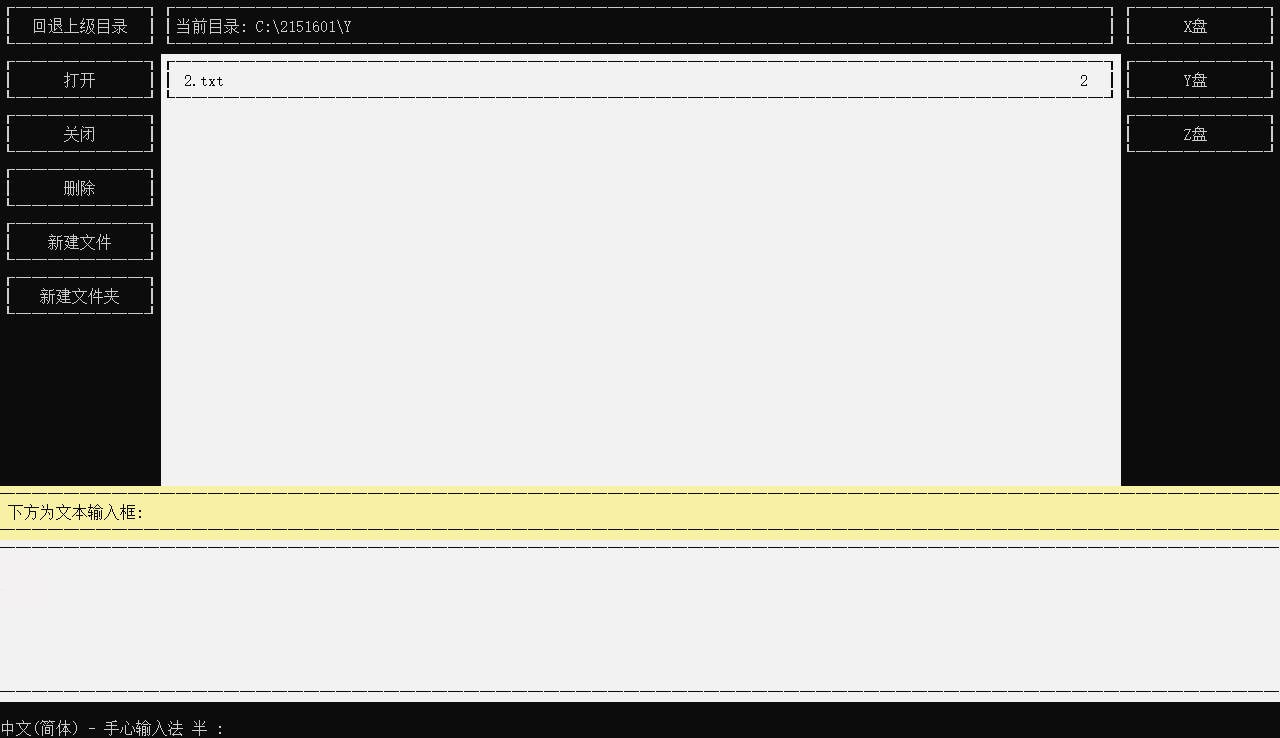
抬头目录可以发现在C:\2151601\Y，同时可以看到该盘中已经成功建立一个1.txt文件。

1. **写文件**

选择一个文件，对其进行打开操作，在下方文本输入框输入123。

1. **读文件**

打开文件，可以看到该文件的内容123被读出来

1. **删文件**

对1.txt执行删除操作可以看到Y盘中只存在2.txt。

1. **在盘中新建文件夹，在文件夹新建文件**

可以看到抬头目录为C:\2151601\X\5。