华 南 农 业 大 学 信 息（软 件） 学 院

《操作系统分析与设计实习》成绩单

开设时间：2023 学年第一学期

<table>
<tr><td colspan="10">小组成员、组内分工、工作量比例、各成员个人成绩</td></tr>
<tr><td>学号</td><td>202125310408</td><td>姓名</td><td>黄海源</td><td>分工</td><td>后端架构设计与代码实现</td><td>工作量比例</td><td>32</td><td>成绩</td><td></td></tr>
<tr><td>学号</td><td>202125310407</td><td>姓名</td><td>何勤政</td><td>分工</td><td>前端界面的设计和功能实现</td><td>工作量比例</td><td>31</td><td>成绩</td><td></td></tr>
<tr><td>学号</td><td>202125310406</td><td>姓名</td><td>古权锋</td><td>分工</td><td>文件窗口和报告撰写</td><td>工作量比例</td><td>18</td><td>成绩</td><td></td></tr>
<tr><td>学号</td><td>202125310405</td><td>姓名</td><td>高晨</td><td>分工</td><td>FAT 表格可视化和报告撰写</td><td>工作量比例</td><td>19</td><td>成绩</td><td></td></tr>
<tr><td>实验题目</td><td colspan="9">模拟磁盘文件系统</td></tr>
<tr><td>各人分工与体会</td><td colspan="9">古权锋：本次课程设计中，我主要负责文件窗口和报告撰写。这次实践让我对 javafx 有了更多的了解，同时我也学会了使用 git-github 来合作开发项目。
黄海源：我主要负责整个后端的架构设计和代码实现。这次实践使我学会了使用 git-github 进行团队项目开发，且学会了预留后端接口供前端衔接。
高晨：这次实践我主要负责 FAT 表的可视化工作，这加深了我对 javafx 的理解，对 git 的运用也更加得心应手。
何勤政：我主要负责前段界面的设计和功能的实现，学会了如何和后端衔接，使用 git-github 合作开发项目，学会使用 git 做版本维护。</td></tr>
<tr><td>教师评语</td><td colspan="9">评价指标：
● 题目内容和要求完成情况 优 □ 良 □ 中 □ 差 □
● 对算法原理的理解程度 优 □ 良 □ 中 □ 差 □
● 程序设计水平 优 □ 良 □ 中 □ 差 □
● 程序运行效果及正确性 优 □ 良 □ 中 □ 差 □
● 课程设计报告结构清晰 优 □ 良 □ 中 □ 差 □
● 报告中总结和分析详尽 优 □ 良 □ 中 □ 差 □</td></tr>
<tr><td></td><td colspan="5"></td><td colspan="2">教师签名</td><td colspan="2"></td></tr>
</table>

一.需求分析

1）支持多级目录结构，支持文件的绝对读路径；

2）文件的逻辑结构采用流式结构，物理结构采用链接结构中的显式链接 方式；

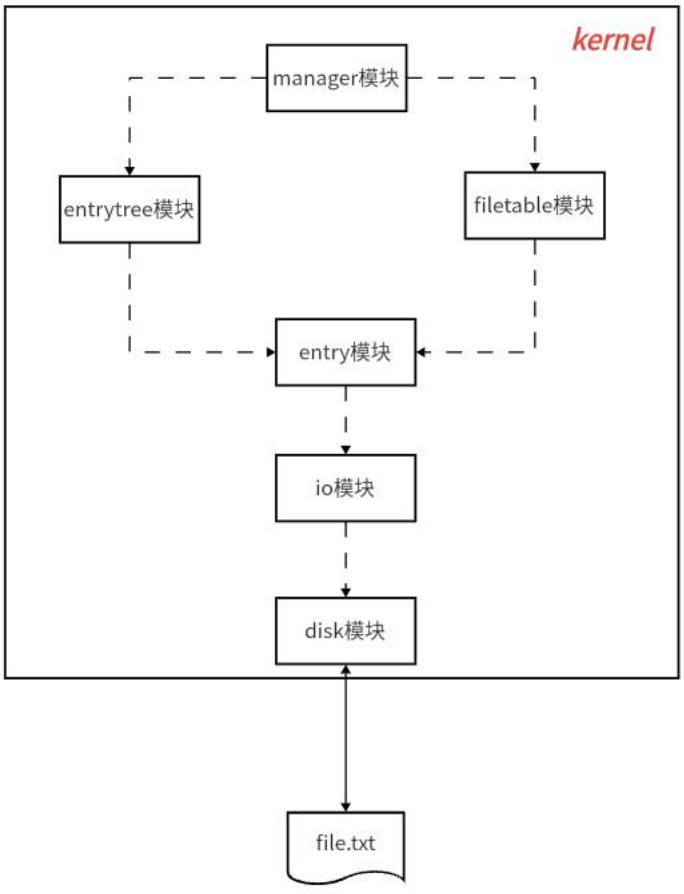
3）采用文件分配表 FAT；

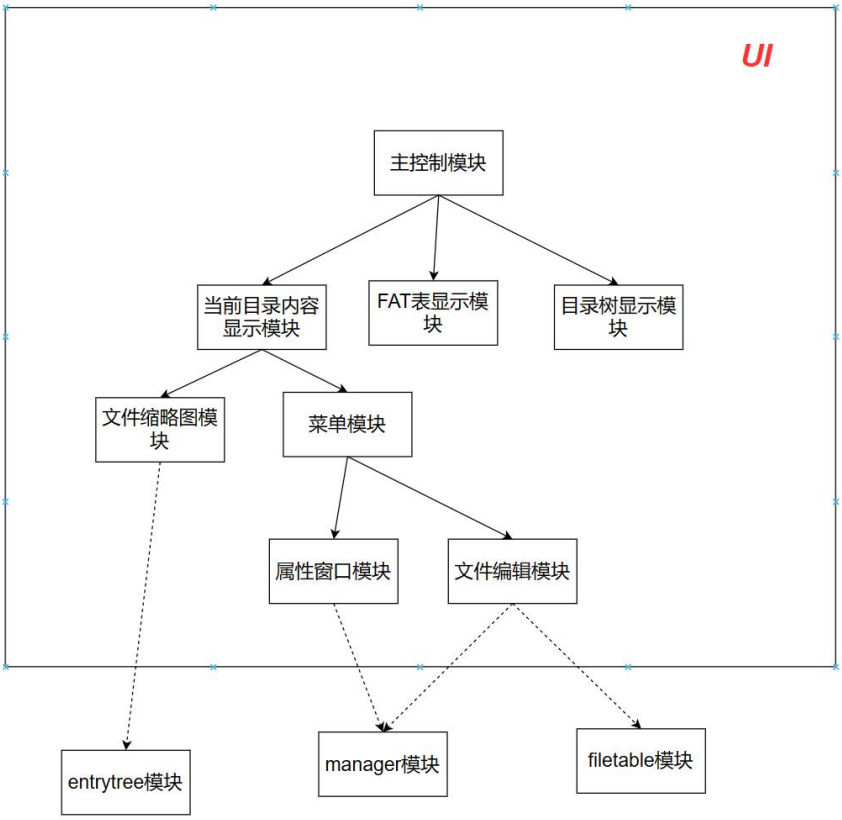
4）实现的命令包括建立目录、列目录、删除空目录、建立文件、删除文件、显示文件内容、打开文件、读文件、写文件、关闭文件、改变文件属性。可以采用命令行界面执行这些命令，也可以采用“右击快捷菜单选择”方式执行命令；

5）最后编写主函数对所作工作进行测试。

二.概要设计

2.1 程序架构概要





2.2 模块功能概要

MainViewController：控制下面模块的建立、调用和刷新，讲各个模块相互关联起来；可视化显示当前目录中所有文件和子文件夹；界面上方当前所处目录文字显示，支持前进、后退等操作。

BlockTable：根据 FAT 表在可视化界面右上角显示各个块占用情况，连续块用同一种颜色表示。

FATTable：根据 FAT 表在可视化界面右下角以两列表格形式显示各个块的索引情况。

MyTreeItem：可视化显示左侧目录树，处理其上的鼠标点击事件，支持展开和隐藏功能。

FILeFlowPane：在窗口中间显示一个文件夹下的所有文件，和菜单(OperationMenu)、文件图标(ThumbnailPane)共同处理这个范围内的鼠标点击事件，支持打开文件夹、打开文件、查看属性、删除文件。

FileWindowMain：展示打开文件的窗口，分为 2 个部分：文件内容展示文本框，展示文件夹当前已有的数据信息；文件内容追加文本框，向文件夹追加内容。点击保存后，会向文件追加内容。

disk：提供一个通过缓存对单个块进行读写的最底层方法。

entry：目录项的抽象，提供目录项层面的方法。

entrytree：目录项作为目录树节点的抽象，提供目录树层面的方法。

filetable：目录项作为文件句柄的抽象，提供文件操作层面的方法。

io：根据 disk 提供的底层读写方法，提供更多的输入输出方法封装方便顶层调用。

manager：提供文件系统层面的方法，作为文件管理系统的功能接口供前端调用。

三. 详细设计

3.1 后端设计

3.1.1 硬盘存储结构

disk：一个文件，分为 128 块，每块 64B，第 0 块和第 1 块存储文件分配表，第 2 块为根目录块用于存储根目录。

FAT：FAT 文件分配表一共有 128 个字节，每一个字节存储对应块的下一块位置。若值为-1 代表结束，0 代表未被使用。由于第 0 和第 1 块与第 2 块分别用于存储 FAT 和根目录，所以将该 FAT 的第 0，1，2 字节值固定为-1，且不得更改。

目录项：目录项大小为 8 个字节，前 3 个字节为目录名，第 3 个字节为扩展名，文件属性存在第 4 字节，接下来的一个字节为起始盘块号，最后两个字节存储文件长度。

3.1.2 输入输出

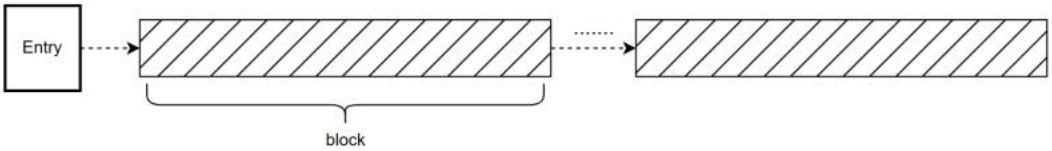
disk 模块中的 Disk 类通过 java 自带的 RandomAccessFile 类实现对 disk.txt 的读写来模拟硬盘读写，其中只提供了一个通过缓存对单个块进行读写的最底层方法。

io 模块中的 IOtool 类则在 disk 提供的最底层方法的基础上衍生出许多 I/O 方法供上层调用，Pointer 类提供了一个遍历文件/目录空间的迭代器供上层调用，封装了遍历时指针字长变化和块位置跳转以及越界检测等细节。

3.1.3 数据结构

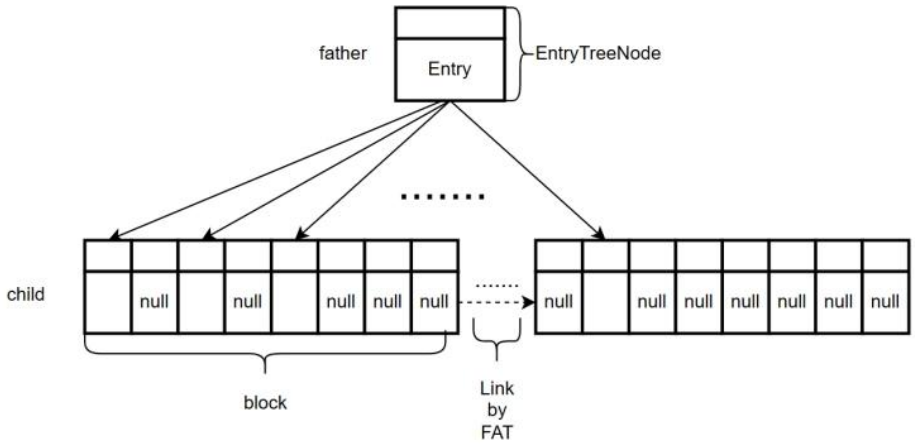
程序启动后，数据从硬盘（disk.txt）中加载到内存，内存中的数据结构可以根据模块分为以下三个层级：

entry 模块是目录项的抽象，确定了以单个块为节点的链表作为文件/目录内容的存储形式，这个链表是根据 FAT 表中的索引进行链接的，如下图所示。尽管目录项可以是目录的目录项或文件的目录项，但 entry 模块偏向于提供目录的操作方法，如在当前目录项中创建、删除、或返回子目录项，因为这些方法同样是面向目录项的，而文件操作方法则交由 filetable 模块提供。值得注意的是，在我们的实现当中并不限制子目录项的数量，若当前目录项的目录空间已满，则会类似文件一样在硬盘中开辟一个新的块，当前目录项的最后两个字节存储目录长度。

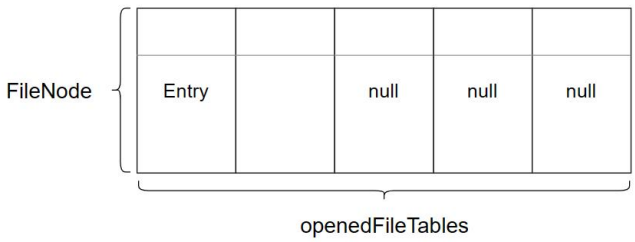


entrytree 模块提供目录树层面的方法，其中的 EntryTreeNode 类是目录项作为目录树节点的抽象。程序启动后，将会从硬盘中加载目录的拓扑关系，并在 entrytree 模块内创建一棵内存当中的目录树，树上的每一个节点绑定一个 Entry 类实例，如下图所示，该模块中的节点也是前端文件缩略图所绑定的对象实例。该模块关注面向整个目录树的操作方法，并屏蔽底层 entry 模块的细节，极大方便了目录的增删查改。

首先，对于增加子目录，当基于某个目录在硬盘中创建子目录时，目录树将同步新增一个对应的节点。其次，对于删除子目录，则从 entrytree 模块出发选择一个待删除节点，遍历其整棵子树，得到所有绑定的 Entry 类实例并全部删除，释放硬盘空间。最后，对于其他查询操作，如查询某目录的子目录数量，或查看当前文件在根目录中的位置，都可以只在 entrytree 模块的层面当中进行。



filetable 模块提供一系列文件操作方法，其中的 FileNode 类是目录项作为文件句柄的抽象，而 FileTableHelper 维护了一个数组 openedFileTables 作为已打开文件表，其中存储 FileNode 实例，表示其对应的文件已被打开，如下图所示，该模块中的节点也是前端文件编辑所绑定的对象实例。对于已打开的文件，则可以继续调用读、写、关闭文件等方法，实现文件操作。



3.1.4 功能接口

manager 模块中的 Manager 类提供了题目要求实现的功能方法，以及一些前端可视化所必要的数据提供方法，作为接口供前端调用。这些方法包括创建、删除、返回子目录项，打开、关闭和读写文件，改变文件属性、返回硬盘使用情况等。

3.2 前端设计

3.2.1 FAT 表可视化

主要由 BlockTable 和 FATTable 子模块实现，其中前者负责根据 FAT 表在可视化界面右上角显示各个块占用情况，连续块用同一种颜色表示。具体来说先建立和初始化 128 个 javafx 的 Rectangle 实体，通过后端提供的接口 listFATBlockLink 得到每个不同的块占用情况和不同连续块的索引，将同一连续块标注为同一颜色。后者根据 FAT 表在可视化界面右下角以两列表格形式显示各个块的索引情况，具体是通过后端得到的 FAT 表数组分成两列显示在表格中

3.2.2 目录树的展示

使用 javafx 提供的 TreeView 类作为基础，myTreeItem 为目录树的每一项。当目录树的一项被选中时会在当前内容模块显示被选中文件夹所包含的文件。当树的一项被展开，myTreeItem 会通过后端目录树对应节点获取最新的子文件夹，更新到前端目录树种。展示该文件夹下的所有子文件夹

3.2.3 前进后退的实现

三个按钮分别对应：上一步、下一步和返回父目录。具体实现时使用 2 个栈（前进栈和后退栈）分别记录上一步和下一步的文件目录。点击上一步时，将后退栈栈顶元素放入前进栈，删除后退栈栈顶元素，打开后退栈新栈顶的文件夹。下一步同理。返回父目录使用 EntryTreeNode 提供的父目录接口，打开父目录，并且清空前进栈。

3.2.4 文件内容显示和编辑

主要使用两个 TextArea 类和一个按钮来实现，两个 TextArea 分别对应显示文件内容和编辑追加内容，按钮对应将追加内容写入文件。点击保存按钮时，将编辑框中的文本追加进文件并在内容展示框中更新。

四.调试分析

4.1 调试过程中遇到的问题是如何解决的以及对设计与实现的讨论和分析

4.1.1 Github 配置问题

这个项目是通过 git+github 协作开发，使用 IDEA 编写，开始合并其他人上传的版本后会导致 IDEA 找不到 SDK，而且运行不了项目，后来通过使用 git 在多个版本来回穿梭

比较发现，IDEA 会在项目根目录自动生成一个配置文件夹.idea，但这个文件夹没有被加到 gitignore 中，这导致合并项目之后使用的是其他人电脑的配置文件，配好 gitignore 之后问题就解决了。

4.1.2 图标显示问题

显示图标的问题，时有时无。某个版本确定能显示图片，但在合并其他分支之后显示不了了，我回滚回原本能显示图标的版本发现也显示不了图标，这就能定位到是配置问题，输出调试和网站搜索，确定是程序运行时没用定位到资源文件夹的位置，使用 maven 配置好项目，就解决问题了。

4.2 设计过程的经验和体会

古权锋：在本次程序设计中，我主要负责编写显示文件内容和对文件进行追加数据的代码以及调试。在刚开始时，由于代码更新不及时以及不理解组员的代码思路和功能，导致我的工作开展的十分困难，在不断的尝试和组员的帮助下慢慢的理解了大部分底层代码的实现和功能。编写过程中，我参考了电脑中文本的操作决定主要使用 TextArea 类来模拟文件的追加，根据文件的类型来决定是否可以对 TextArea 类进行操作。

高晨:在本次程序设计中，我主要负责编写块占用可视化和 FAT 表格显示功能模块。在刚开始时，由于对文件结构知识的理解不到位导致难以下手，而后在组员和网络的帮助下掌握了相关知识。在编写过程中，考虑到块占用情况的易读性，对连续的块使用同一种颜色表示。

何勤政：在本次程序设计中，我主要负责主窗口有关模块，以及和模拟磁盘后端对接。由于之前其他可的程序设计实验也用到了 javafx，很顺利就完成了主体框架的搭建，比较困难的调试，由于模拟磁盘并不是自己写的，所以需要先看懂代码，然后通过输出过程，输出调用栈，来定位问题的地方，并及时反馈。由于$和#是磁盘的关键字符，而且磁盘对文件名、扩展名有限制，所以在前端给文件、文件夹命名的时候要使用正则表达式限制用户的输入，防止用户输入非法字符。为了让程序有更好的鲁棒性，在用户操作的时候最大限度的确保了用户的非法操作，保证非法操作不会到达后端。

黄海源：在本次程序设计中，我主要负责整个后端的架构搭建和代码实现。在搭建框架之前，我反复研读题目，明确题目要求我们做到硬盘读写级别层面的模拟，因此，我单独创建一个 Disk 类，仅提供通过缓存对单个块进行读写的最底层方法，接着根据这个方法构筑整个后端的输入输出方法集，并设计一个迭代器封装底层的读写遍历，方便顶层的数据结构调用。在数据结构的设计上，我实现了 1）目录项本身的抽象、2）目录项作为目录树节点的抽象以及 3）目录项作为文件句柄的抽象三个层面的分离。第 1 点关注对单个目录的操作，涉及的数据结构是根据 FAT 表链接的块链表，这里我没有限制目录仅占一个块

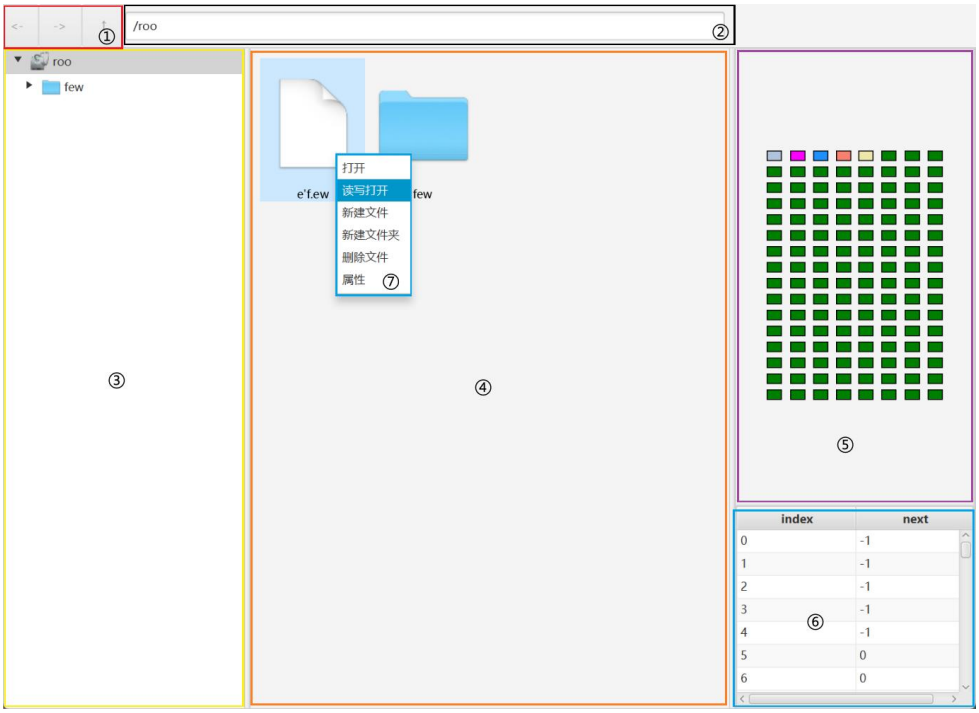
的空间，而是与文件一样，在空间不足时开辟新的块，因此可以在硬盘空间的允许下创建任意多个目录项。第 2 点关注整个目录树层面的操作，尤其对于删除目录操作，需要从目录树层面定位到待删除节点后，通过遍历子树获取所有的 Entry 实例，再执行所有 Entry 实例中遍历整个链表的删除操作并释放空间，整个操作串通两个数据结构，一气呵成。由此也看出，我所实现的删除目录操作不仅局限于题目要求的删除空目录操作。第 3 点关注文件操作，我在设计方法时考虑得比较底层，尽可能预留多的接口，类似 C 语言中得到文件句柄后进行读、写文件操作时需要指定指针位置和读写数量，且读写后指针不会复位，因此也造成实现前端的同学在调用后端方法时因为不理解逻辑而出现 bug，这也说明在团队合作时需要多留注释并加强沟通，以避免信息不对称所造成的时间和精力上的浪费。

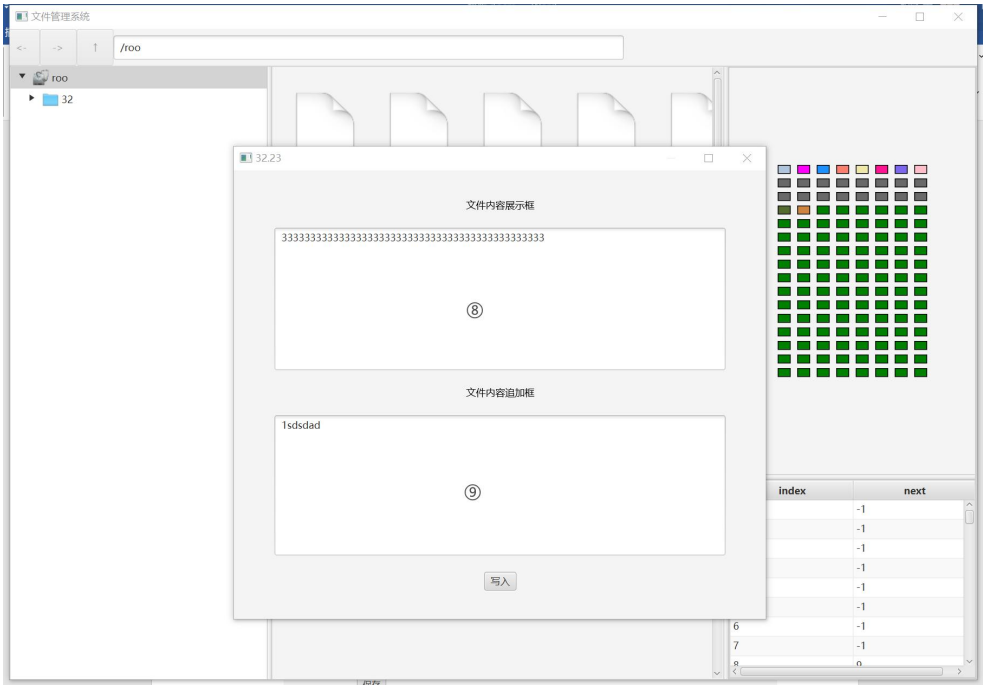
4.3 实现过程中出现的主要问题及解决办法

在本次程序设计中，遇到的问题有无法多次追加内容进入文件，在调试中我们发现是在读取文件内容之后，指针没有复位再次读取文件内容导致越界，后续我们直接对 TextArea 类进行操作从而避免对文件内容进行多次读取而导致越界。

五.用户使用说明

鼠标左键双击左侧目录树文件夹可进入该文件目录并展开其子目录；双击文件夹可进入该目录，双击文件可查看文件内容进行编辑；鼠标右键单击文件或文件可选择打开、读写打开、属性、新建文件、新建文件夹、删除文件（文件夹）；右边可实时预览块占用情况和 FAT 表，其中右上块占用可视化中同一颜色表示是被连续占用，右下 FAT 表左列是区块索引，右列是下一个区块的地址。





① 回退，上一步和下一步

② 显示文件绝对路径

③ 文件目录树

④ 当前目录内容

⑤ 磁盘占用情况

⑥ FAT 表

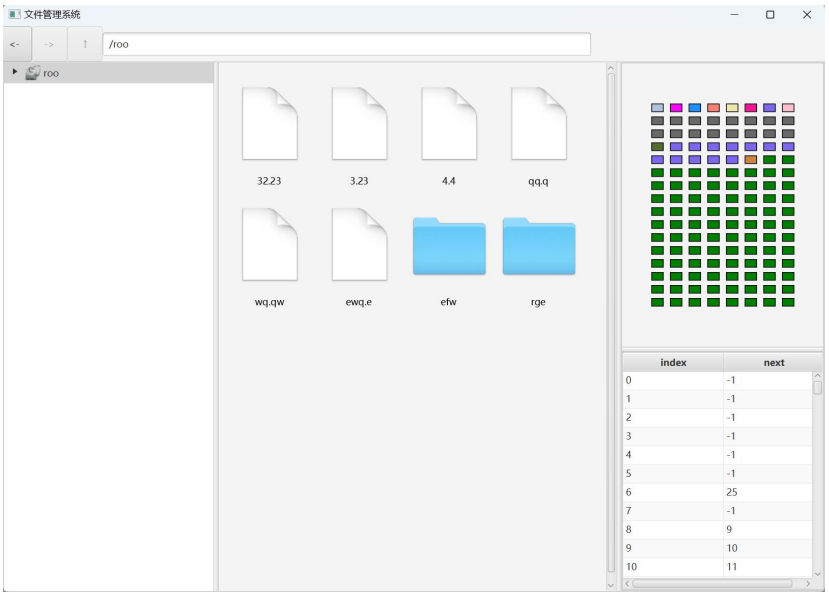
⑦ 右键目录菜单

⑧ 显示文件内容

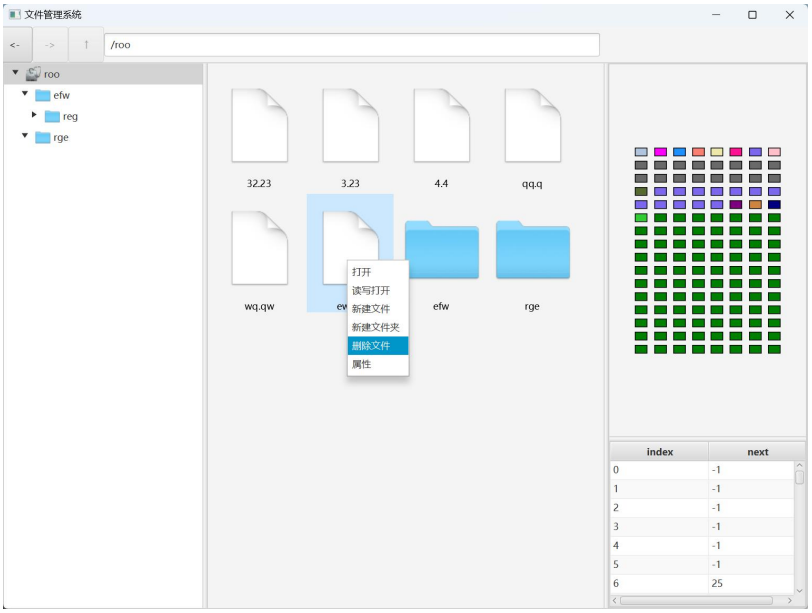
⑨ 追加文件内容

六.测试与运行结果

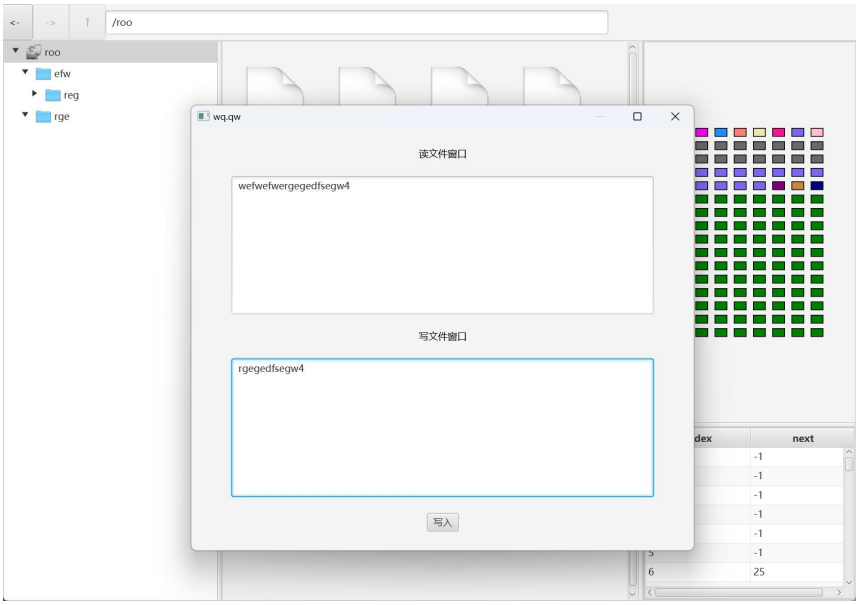
6.1 系统主界面



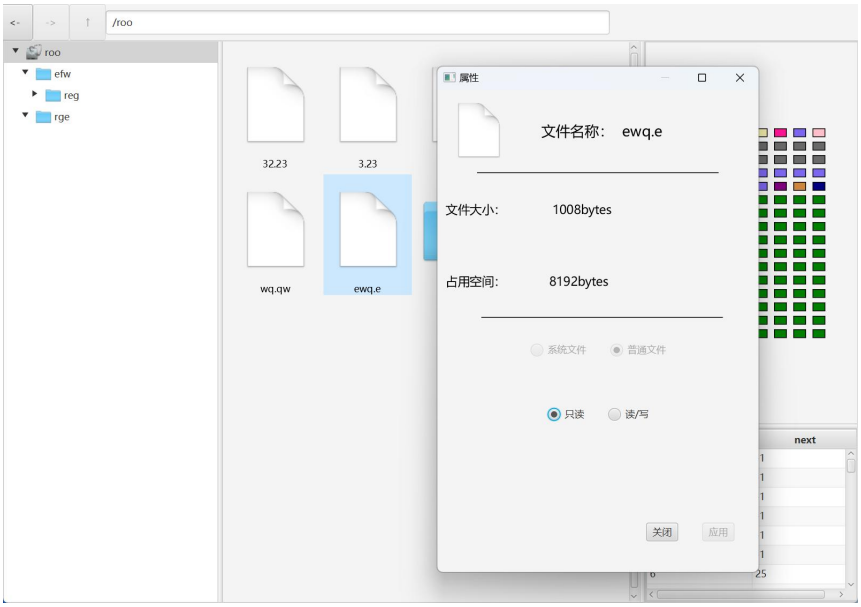
6.2 右键菜单栏功能测试



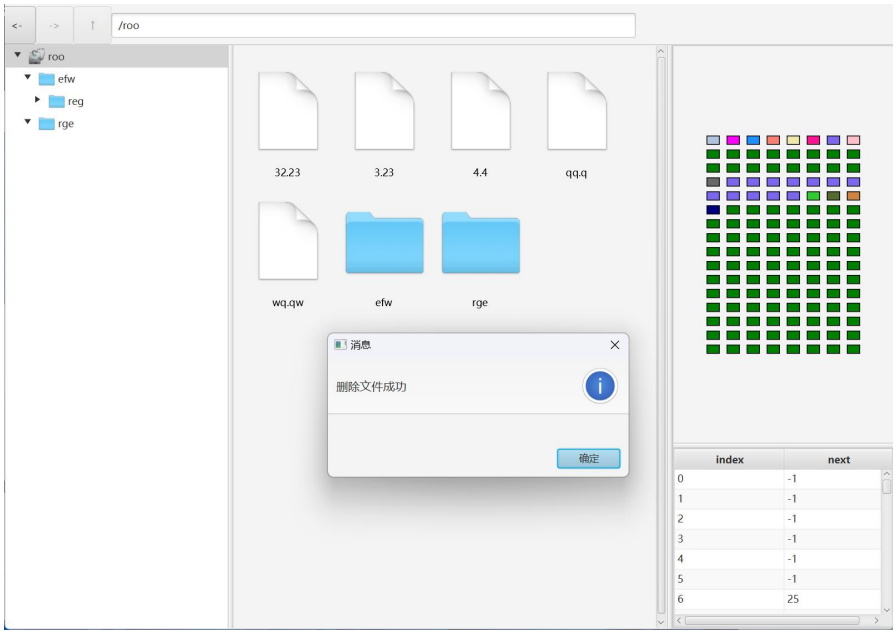
6.3 打开文件、文件夹


/roo
roo
efw
reg
rge
wq.qw
读文件窗口
wefwefwergegedfsegw4
写文件窗口
rgegedfsegw4
写入
next

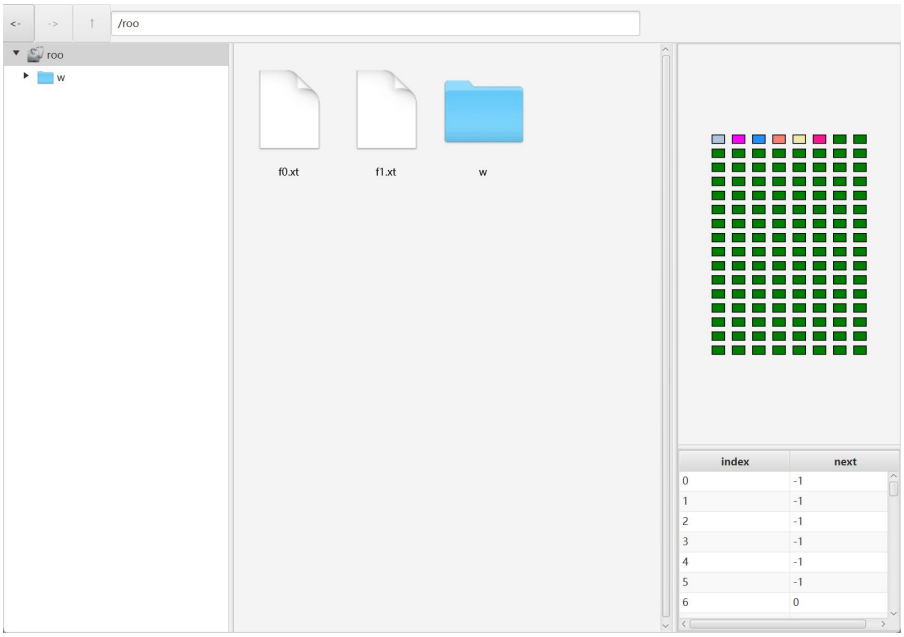
6.4 文件属性


/roo
roo
efw
reg
rge
32.23
3.23
wq.qw
ewq.e
属性
文件名称： ewq.e
文件大小: 1008bytes
占用空间: 8192bytes
系统文件
普通文件
只读
读/写
关闭
应用
next

6.5 删除文件



6.6 新建文件和文件夹，文件分配表、磁盘使用情况与预期符合



6.7 删除文件夹,文件分配表相应项清空，磁盘使用情况更新