



Tongji University

操作系统课程项目实验报告

003-文件系统

专 业： 软 件 工 程

指导教师： 张 惠 娟

学 号： 2 1 5 3 0 6 1

姓 名： 谢 嘉 麒

1. 项目及基本要求概述

1.1. 项目概览

1.1.1. 项目目的

- (1) 理解文件存储空间的管理；
- (2) 掌握文件的物理结构、目录结构和文件操作；
- (3) 实现简单文件系统管理；
- (4) 加深文件系统实现过程的理解；

1.1.2. 项目要求

- (1) 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；
- (2) 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) 文件存储空间管理可采取链接结构（如 FAT 文件系统显式链接等）或者其他学过的方法；
- (4) 空闲空间管理可采用位图或者其他方法；
- (5) 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。
- (6) 文件系统提供的操作：
 - A. 格式化
 - B. 创建子目录
 - C. 删除子目录
 - D. 显示目录
 - E. 更改当前目录
 - F.

1.2. 项目环境

1.2.1. 编程语言

本次项目考虑到需要进行可视化实现，同时也要实现进一步的用户交互和虚拟内存管理以及日志记录等功能，故选择 C# 语言作为项目编程语言。

1.2.2. 操作系统平台

本次项目使用 Windows 系统作为操作系统平台，所有程序均在 Windows10 系统下运行。

1.2.3. 项目编辑及运行方法

本次项目选择 Visual Studio2022 作为编辑器，其中 VS 需要安装 C# 窗体应用开发的相关组件（安装来源 Visual Studio Installer）

项目运行需要用 VS2022 打开项目源代码文件夹下的 .sln 文件，选择 Debug 或 Release 模式进行运行，请确保文件完整。

项目运行也可以选择直接打开可执行文件文件夹下的 .exe 文件，请确保文件完整。

2. 整体设计思路

本次项目整体分为四个部分，包括虚拟内存处理，目录及文件结点设置，文件系统可视化及操作和本地日志存储。

2.1. 虚拟内存处理

本次项目在内存中开辟一个空间作为虚拟文件存储器，并对此空间做如下处理：

2.1.1. 虚拟内存属性设置

在虚拟内存类中设置磁盘容量，剩余磁盘块数，块大小，磁盘块数，内存空间和位图几个属性，并在程序开始时为这几个属性赋初值。

其中磁盘容量和块大小可以直接规定，并利用 $\text{磁盘块数} = \text{磁盘容量} / \text{块大小}$ 来计算磁盘块数。为内存和位图申请空间后，将位图全部初始化为 -1，表示全部可用；将内存全部初始化为 " "，表示内存全空。

2.1.2. 文件存储空间管理

文件存储空间采用 FAT 文件系统中的显式链接。文件中的内容存放在磁盘中的不同块，这些块之间利用位图进行指引和链接。当文件创建时，如果需要申请内存，则根据申请的内存大小为文件分配一定数量的空闲块。写文件时不断查找空闲块并写入，更新索引。删除文件则直接置空原有内存位置。

2.1.3. 空闲空间管理

空闲空间利用位图进行标识。位图与内存块存在一一对应关系，当某一位置位图标识为 -1

时，对应块的内存是空闲内存；同样的，当某一位置位图标识不为-1，则表示此位的对应盘块有内容，且位图中的数字指向存放此文件内容的下一盘块的位置；当位图标识为-2时，表示文件结束。

2.2. 文件及目录设置

基础文件信息主要包括文件名，文件类型（txt 文件或文件夹），文件最近一次修改时间，文件大小和文件在内存中存放的初始位置。

目录采用树的二叉树表示法，结点类在定义文件结点的基础上添加左孩子，右兄弟指针，并添加指针指向父母结点。

为了照顾文件系统的具体操作，需要在目录中添加包括删除根节点，搜索节点，添加文件（夹），删除文件（夹）和找到根节点等功能。

2.3. 文件系统操作

文件系统参考 Windows10 的文件系统，主要包括以下几个部分：

2.3.1. 文件系统界面

文件系统界面与 Windows10 文件界面相似，主要包括左侧目录树，上方文件路径显示和中部文件结构及细节信息展示。文件界面下方会显示当前磁盘总容量和剩余磁盘容量。

2.3.2. 新建文件/文件夹操作

在文件界面最上方点击新建文件/新建文件夹按钮来新建，系统会弹窗提示输入文件名/文件夹名并自动将其存储到目录中。

2.3.3. 返回上一级目录

在文件界面最上方点击返回上级目录按钮可以返回上一级目录。

2.3.4. 格式化

在文件界面最上方点击格式化按钮，会清空当前目录，清空内存和位图内容，初始化整个系统。

2.3.5. 文件细节操作

在文件界面中部文件细节信息中，可以通过鼠标右键单击某一行来获得操作栏目，栏目中包括打开文件/文件夹，删除文件/文件夹和重命名文件/文件夹功能，可以分别点击使用。

2.3.6. 目录树操作

目录树会显示目录结构，用户也可以通过双击目录树中某一项来跳转到相应的路径。

2.4. 本地日志存储

由于项目要求实现系统关闭后再打开时继承上一次的信息，则需要将信息存储在本地的日志文件中。

系统在本地运行目录下自动建立 info 文件夹存放日志，日志主要包括 catalog.txt, BitMap.txt 和 Disk.txt，分别存放目录信息，位图信息和磁盘信息。

系统在关机后会自动将信息存放在以上文件中，下一次启动时会自动读取，解码并加载信息，实现信息的回复功能。

3. 系统各功能实现

3.1. 虚拟内存各功能实现

3.1.1. 读取文件内容功能

读取文件内容包括几个步骤：

- (1) 根据文件节点得到内容在内存中的盘块初始序号；
- (2) 读取内存盘块；
- (3) 根据位图找到下一块盘块
- (4) 重复 (2) (3) 直到读取所有内容，即可结束，返回读到的内容。

3.1.2. 删除文件内容功能

删除文件内容和读取文件内容相似，只需要不断清空内存盘块并依托位图找到下一块即可，需要额外更新磁盘剩余容量这一属性。

3.1.3. 为新文件分配空间

为新文件分配空间：

- (1) 首先根据内容计算需要占用的盘块数；
- (2) 如果内存足够，则进入下一步，否则报错退出；
- (3) 遍历位图找到空闲位置，将这一位置作为起始位置，存放盘块大小的，更新标志；
- (4) 不断找到下一个空闲的位置，更新位图，存放内容，直到存储整个文件。
- (5) 整个过程需要不断更新磁盘剩余容量。

3.2. 目录及文件各功能实现

3.2.1. 删除目录功能

根据目录的根节点来递归删除某个目录，主要用于用户删除文件夹时进行操作。

- (1) 函数返回条件是当前指针遍历到叶节点；
- (2) 首先清空左孩子，不断递归调用，传入当前指针的左孩子，最后清空根节点的左孩子；
- (3) 之后清空右兄弟，不断递归调用，传入当前指针的右兄弟，最后清空根节点的右兄弟；
- (4) 最后清空根节点。

3.2.2. 搜索节点功能

用户点击某个项目后可以利用窗体控件得到这一项目的名称信息，需要根据这一名称信息来获得其对应节点。为了防止重名发生，额外传入文件类型和文件根节点作为标识。

- (1) 函数退出条件为指针遍历到叶节点或者找到目标节点或者未找到目标节点；
- (2) 首先不断递归搜索左孩子节点；
- (3) 未找到则递归搜索右兄弟；

3.2.3. 添加文件/文件夹

本功能为文件系统基础功能。

- (1) 如果没有根节点就直接返回；
- (2) 根据新文件名/文件夹名找到父母节点；
- (3) 如果父母节点左孩子空，则直接加入；
- (4) 否则设置临时节点，辅助遍历右兄弟直到找到插入位置；
- (5) 找到位置后加入。

3.2.4. 删除文件/文件夹

本功能为文件系统基础功能。

- (1) 根据文件名搜索定位文件节点；
- (2) 找到其父母节点
- (3) 如果它是父母节点的左孩子节点，则直接删除；
- (4) 如果它是父母节点的右兄弟节点，则执行以下部分：
 - A. 设置临时节点暂存，辅助遍历右兄弟节点，寻找删除位置；
 - B. 找到位置后，清空节点。

3.2.5. 找到当前节点的最近根节点

这一功能的设置主要是辅助日志信息的存储。在日志信息中需要存储节点最近的父母节点，利用这一函数进行搜索。

- (1) 自身或左孩子为空，则直接返回；
- (2) 否则如果找到后直接返回找到的节点；

(3) 设置临时节点，辅助向右兄弟方向遍历；

(4) 在左孩子中递归查找；

(5) 在右兄弟中递归查找；

3.3. 文件系统操作

在文件系统中设置目录，根节点，虚拟内存和当前路径几个变量；

文件系统操作的实现主要依托于各个功能函数的调用和 C#窗体程序的控件设置。

主要包括：

(1) 格式化、返回上级目录、新建文件、文件文件夹四个按钮点击函数；

(2) 目录树的呈现和双击函数；

(3) 主体信息呈现和交互；

(4) 编辑文件的页面；

3.3.1. 按钮点击函数

按钮点击函数只需要为按钮添加点击监听器，并调用相关操作函数即可，不再赘述。

3.3.2. 目录树的呈现和双击

目录树呈现在 `treeView` 控件中，利用 `TreeNode tn = new TreeNode();` 添加节点，利用 `tn.Name`、`tn.Text` 和 `tn.Tag` 添加节点属性。利用递归方法遍历目录，逐个添加进目录树中。

目录树双击需要为控件添加双击监听器，用 `TreeNode tn = FolderTreeView.SelectedNode;` 获取当前点击的节点信息。如果是文件夹，则利用栈结构辅助记录路径并更新，刷新界面到相关文件夹下；如果是文件，则调用文件编辑界面，打开新窗口进行文件编辑。

3.3.3. 主体信息呈现和交互

主体信息呈现当前文件目录下的文件细节结构，包括文件名，文件最近编辑时间，文件类型和文件大小。

主体信息呈现在 `ListView` 控件中。当目录为 `root` 时需要禁用返回上级按钮。按照左孩子右兄弟的结构呈现该目录下的文件/文件夹。利用 `ListViewItem file = new ListViewItem();` 和 `DetailListView.Items.Add(file);` 来为控件添加显示元素。

主体信息交互主要包括鼠标左键和右键点击交互。

利用 `ListViewItem clickedItem = DetailListView.GetItemAt(e.X, e.Y);` 获取当前点击的控件。鼠标右键点击 `ListViewItem` 后，在其右侧显示小栏目，栏目中包括“打开”，“删除”和“重命名”。控件信息会存储到 `Tag` 中，栏目点击后会传递到点击后函数中。点击后函数分别实现文件/文件夹打开，文件/文件夹删除和文件/文件夹重命名三个功能。

鼠标左键点击 `ListViewItem` 后，如果控件是文件，则直接打开文件，跳转到文件编辑界面；

如果控件是文件夹，则跳转到该文件夹下的界面窗口。

3.3.4. 编辑文件界面

用户在目录树中双击文件或在文件信息中打开文件，都会跳转到文件编辑界面。文件编辑界面即记事本窗口，会显示当前文件名和输入文本框。界面内部设置主框架引用，文件名和是否编辑三个变量。

退出时，如果文件未改动，则直接退出；如果文件改动，则弹窗是否保存。

3.4. 日志

日志主要分为记录和读取两个部分。

3.4.1. 日志记录

日志记录时会调用 C# 的文件流函数。将目录（分为父母节点名，节点名，文件类型，编辑时间，文件大小，文件初始位置和结束标识组成），位图和磁盘信息存储在 info 目录下的三个 .txt 文件中。

3.4.2. 日志读取

同样利用文件流函数分别解码并读取三个 .txt 文件中的内容，用来初始化系统。

4. 心得体会

通过本次项目，我学习并掌握了 C# 窗体程序的开发，同时也加深了对文件系统的实现和其本质内核的理解。

通过申请虚拟内存，规定各个内存操作，我进一步学习了显示链接和位图相关知识，熟练掌握了信息在内存中的存储和处理。

通过文件系统的实现，我更加熟悉文件系统的功能和需求，也对文件的存储以及实现有了更进一步的认识。

在日志存储方面，对于日志的写，读和解码操作都为信息的本地存储提供新思路。

5. 运行截图

5.1. 项目开启界面

以下为项目读取 info 内容后的开启界面。



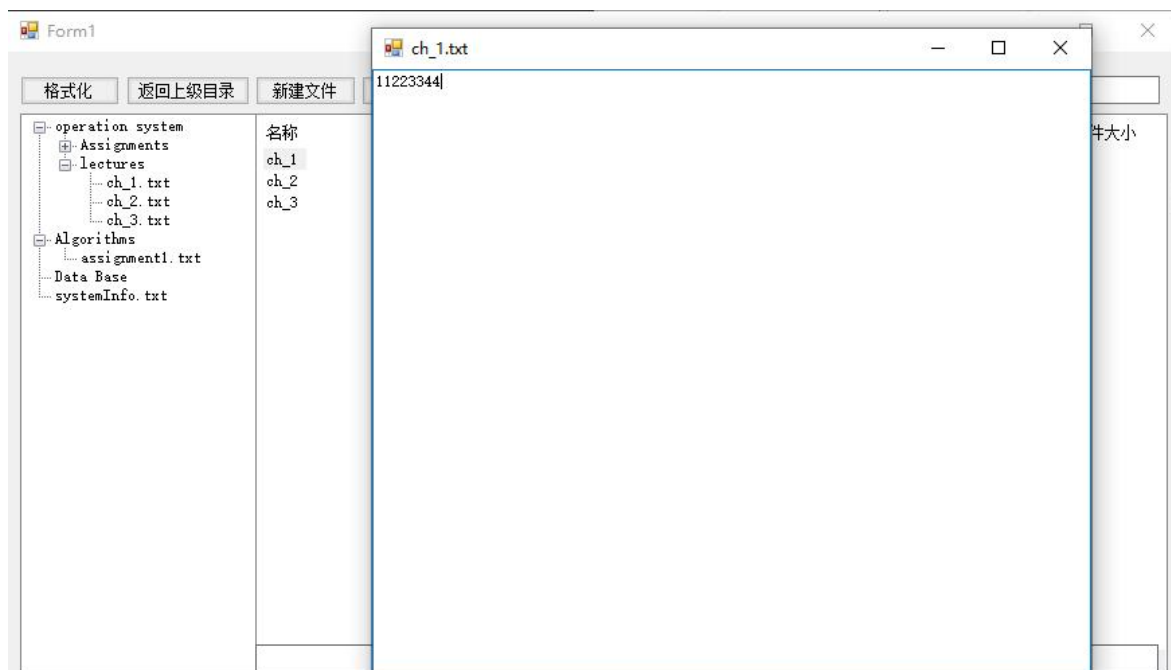
5.2. 更新文件界面



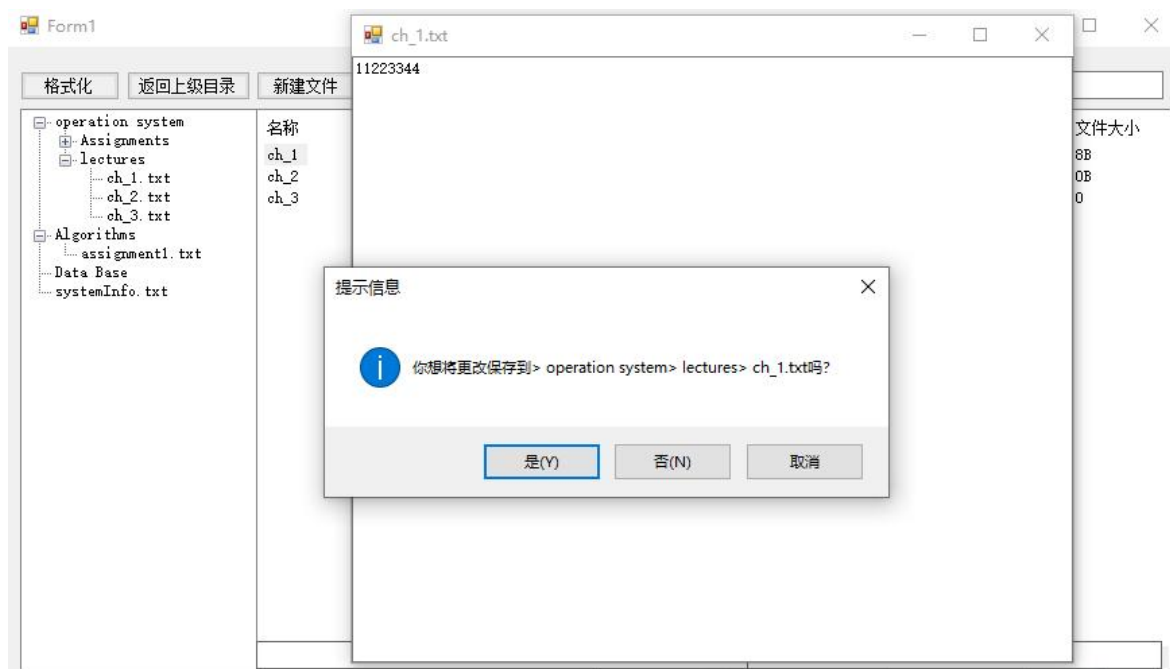
5.3. 文件信息交互



5. 4. 文件编辑界面



5. 5. 文件编辑退出



5.6. 文件重名提示

