******

**文件系统实验报告**

姓 名：曹晓慈

学 号：2052844

所在院系：软件学院

学科专业：软件工程

指导教师：王冬青

二〇二二年六月

**调页存储管理实验报告**

1. **项目介绍**

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，以便下次可以将其恢复到内存中来。

文件存储空间管理采用显式链接，空闲空间管理采用位图。文件目录采用多级目录结构。目录项目中包含：文件名、物理地址、长度、文件类型、修改日期等信息。

文件系统提供的操作包括以下几种：

格式化

创建子目录

删除子目录

显示目录

更改当前目录

创建文件

打开文件

关闭文件

写文件

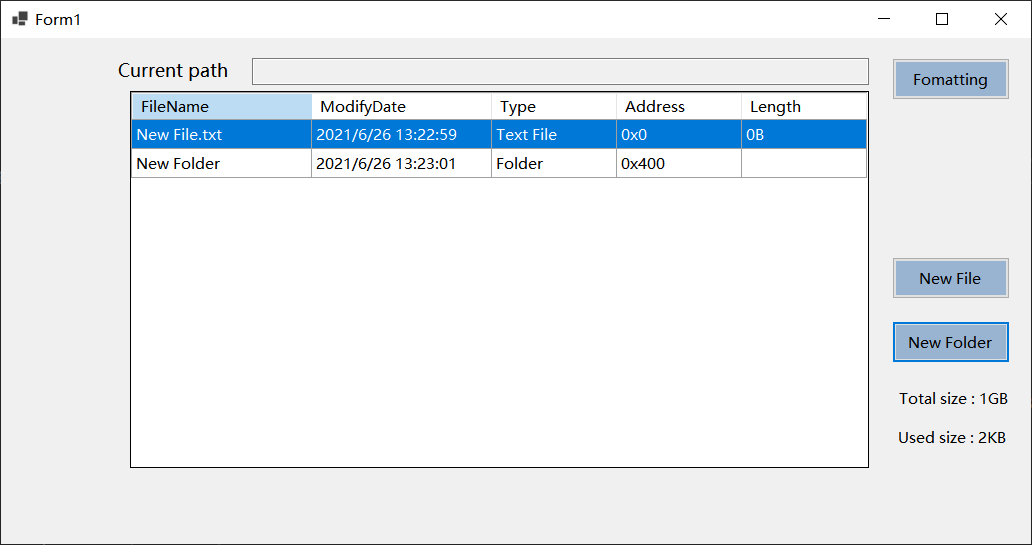
读文件

删除文件

1. **开发环境**

语言为c#，IDE为visual studio

1. **GUI基本构成**



最上方的是当前路径，中间的表格是当前路径下的文件目录。右边三个按钮从上往下依次是：格式化、新建文本文件、新建文件夹。在下面的是内存空间的总大小和当前已使用的空间。同时，可以选中某个文件或文件夹，右键选择打开文件、删除文件或者重命名文件。每次创建文件/文件夹都会先开辟1kb的内存空间，当文件或文件夹大小超过1kb时就去新开辟一个1kb的内存空间。

1. **代码说明**

public class FileNode

{

public string FileName { get; set; }

public string ModifyDate { get; set; }

public bool IsFolder { get; set; }

public int Address { get; set; }

public string content { get; set; }

public FileNode next { get; set; }

public FileNode child { get; set; }

public FileNode parent { get; set; }

public FileNode(string name,string modifyDate,bool isFolder, int address)

{

FileName = name;

ModifyDate = modifyDate;

IsFolder = isFolder;

Address = address;

content = "";

}

}

文件结点类，除了一些基本信息如文件名、物理地址、大小等，整个文件系统其实是一个二叉树，每一层即是一个文件目录，在项目中采用左结点指向子女节点（即child节点），右节点指向兄弟节点（即next节点）的方式。此外，还添加了一个父亲节点（即parent节点），便于查找到当前节点的父节点。

private void button1\_Click(object sender, EventArgs e)

{

int index = dataGridView1.Rows.Add(), address = bitMap.search\_Address();

if (address >= 1024 \* 1024)//内存已满

{

MessageBox.Show("Memory is full!");

return;

}

int newFileIndex = NewFileIndex(false);

//在表格上显示新文件的各个属性

if (newFileIndex == 0)

dataGridView1.Rows[index].Cells[0].Value = "New File.txt";

else

dataGridView1.Rows[index].Cells[0].Value = "New File(" + newFileIndex + ").txt";

dataGridView1.Rows[index].Cells[1].Value = DateTime.Now.ToString();

dataGridView1.Rows[index].Cells[2].Value = "Text File";

dataGridView1.Rows[index].Cells[3].Value = "0x" + Convert.ToString(address, 16);

dataGridView1.Rows[index].Cells[4].Value = "0B";

FileNode newNode = new FileNode((string)dataGridView1.Rows[index].Cells[0].Value,

(string)dataGridView1.Rows[index].Cells[1].Value, false, address);

curNode.next = newNode;//将新文件结点链到当前文件列表中

curNode = curNode.next;

bitMap.Set(address / 1024, true);//更新位图

label3.Text = "Used size : " + (address / 1024+1) + "KB";//更新已用空间

}

Button1是新建文件按钮，点击后在页面上显示新文件的各个属性，并将其添加到当前目录中，同时更新位图。新建文件夹也是类似，此处不再赘述。

private void OpenFile(int index)//index为当前表格中选择的行的下标

{

if (index >= 0)

{

//通过文件名获得当前节点

FileNode Selected = GetFileNode((string)dataGridView1.Rows[index].Cells[0].Value);

if (!Selected.IsFolder)//文件

{

ShowInputDialog(Selected);//展示输入框

//更新保存后的显示的文件大小和修改日期

dataGridView1.Rows[index].Cells[4].Value = Selected.content.Length + "B";

dataGridView1.Rows[index].Cells[1].Value = Selected.ModifyDate;

}

else//文件夹

{

//更新curNode为当前节点子节点

curNode = Selected.child;

curNode.parent = Selected;

//存储当前目录

stack.Push(curCatalog);

curCatalog = curNode;

button4.Visible = true;//显示回退按钮

dataGridView1.Rows.Clear();

textBox1.Text += ">" + Selected.FileName;//更新当前路径

DisplayFileList();//显示当前目录的文件列表

}

}

}

上述函数为打开文件/文件夹的函数，如果是文件，则进入记事本模式，保存用户输入的内容并更新节点相关信息，如果是文件夹，则进入文件夹的目录，即当前节点的子节点中，并对界面中相关信息进行更新。

1. **心得体会**

通过这次文件系统模拟，我对文件存储空间的管理有了更深刻的理解，对于文件的物理结构、目录结构和文件操作，文件系统管理实现等也更加熟悉对文件系统内部功能和实现过程也有了进一步的了解。项目也有很多可以改进的地方，例如，每次新建文件时给文件/文件夹分配1kb是否合理，分配的空间太小就会需要频繁的开拓新空间，这样就会导致查看文件时要调用多个内存块进行查看，分配的空间太大就会导致大量内碎片的产生，使内存空间利用率大大降低，如何给新文件分配合适的空间，也是一个值得思考的问题。另外，也可以给文件添加索引使查找文件更加高效，给哪些文件添加索引，如何添加索引，添加怎样的索引，都是需要去探讨的。总的来说，这次项目带给我的收获不少，但是项目本身也有值得改进的地方。