操作系统文件管理项目

说明文档

1452716

张尹嘉

一、项目分析

1.要求

(1)、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，以便下次可以将其恢复到内存中来。

(2)、具体技术细节：

文件存储空间管理可采取显式链接或者其他方法。空闲空间管理可采用位图或者其他方法。如果采用了位图，可将位图和 FAT 表合二为一。文件目录采用多级目录结构。至于是否采用索引节点结构，自选。目录项目中应包含：文件名、物理地址、长度等信息。同学可在这里增加一些其他信息。

(3)、文件系统提供的操作：

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

2.要求分析：

根据要求，在内存中开辟一块空间作为文件存储器，在程序退出之后应该把文件保存到本地。等到下次打开时应该能够还原上一次的文件目录。因此，我决定用全局文件控制块和本地文件控制块来记录文件之间的关系。同时我在代码中定义类似于文件存储块的数据结构，初始化一定数量的文件存储块作为模拟磁盘。文件控制块中记录本文件的子文件或所占用的控制块，这取决于文件是文件夹类型还是文本文件类型。

同时，程序中还应该提供更改文件目录，新建，删除和格式化的功能。另外，还因该有文

件重命名的功能。

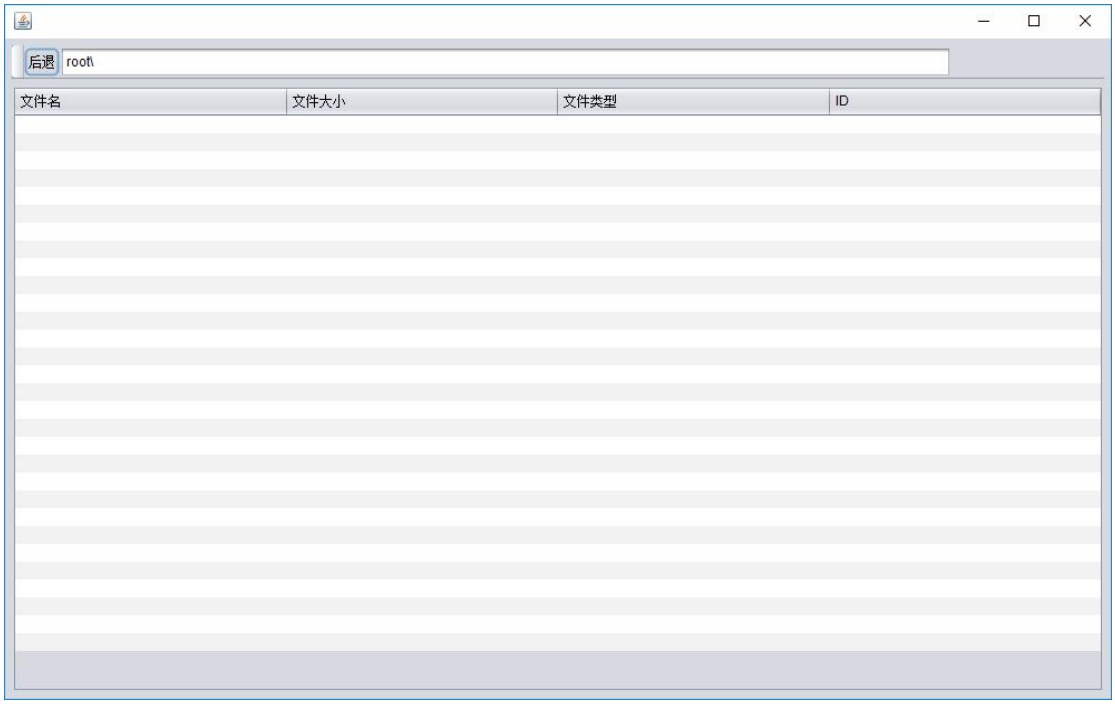
二、程序环境：

本程序使用 JAVA 语言编写，借助了 NetBeans 平台和 Windows 系统。开发时使用的 JDK 版本是 JDK8。但最终生成了 JDK7 和 JDK8 两个版本，以提供兼容性。源代码保存在 SourceCode 文件中，以.java 文件的形式。

三、使用说明：

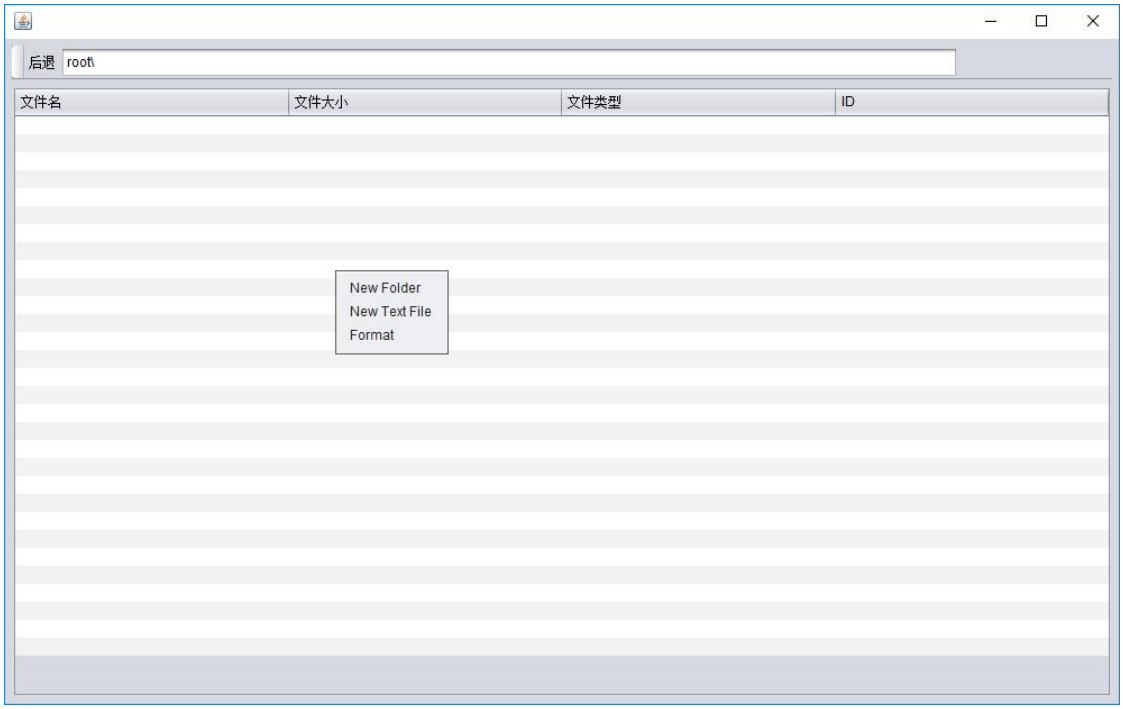
程序由一个 jar 文件和 disk 文件夹组成。jar 文件是可执行文件，disk 文件夹是保存在本地的 FCB 信息和文件存储块信息。请不要修改 disk 文件的内容，否则程序运行可能出现错误，或者发生文件丢失的情况。

打开程序后，显示的是整个文件管理系统的根目录。如下图：

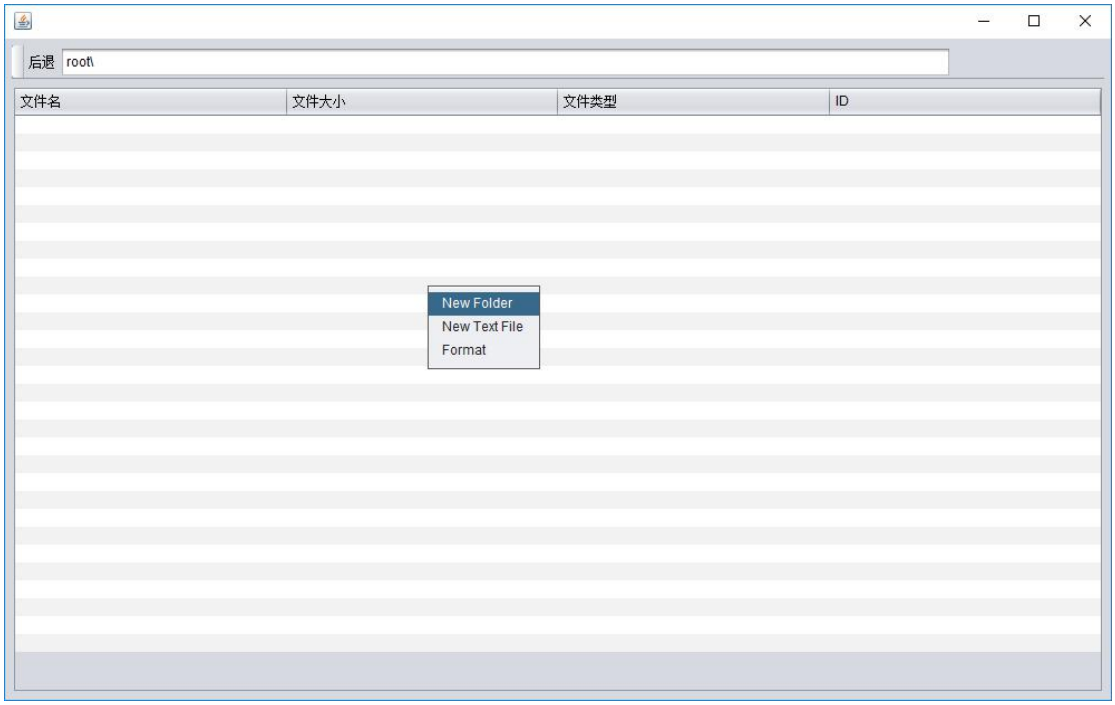


程序的最上端是功能区，后退按钮可以转到当前目录的父级目录上。紧接着是当前路径的文本框，显示了当前路径。程序主体显示了当前目录下的文件和文件夹。文件大小显示的是文件占用的文件存储块的个数。每个文件存储块的最大值是 1kb。该文件管理系统最多支持 1024 个文件存储块的大小。文件类型有 folder 和 text 两种，folder 代表该文件是文件夹，而 text 代表了该文件是文本文件。ID 是文件系统用于唯一标识该文件的 ID 号。

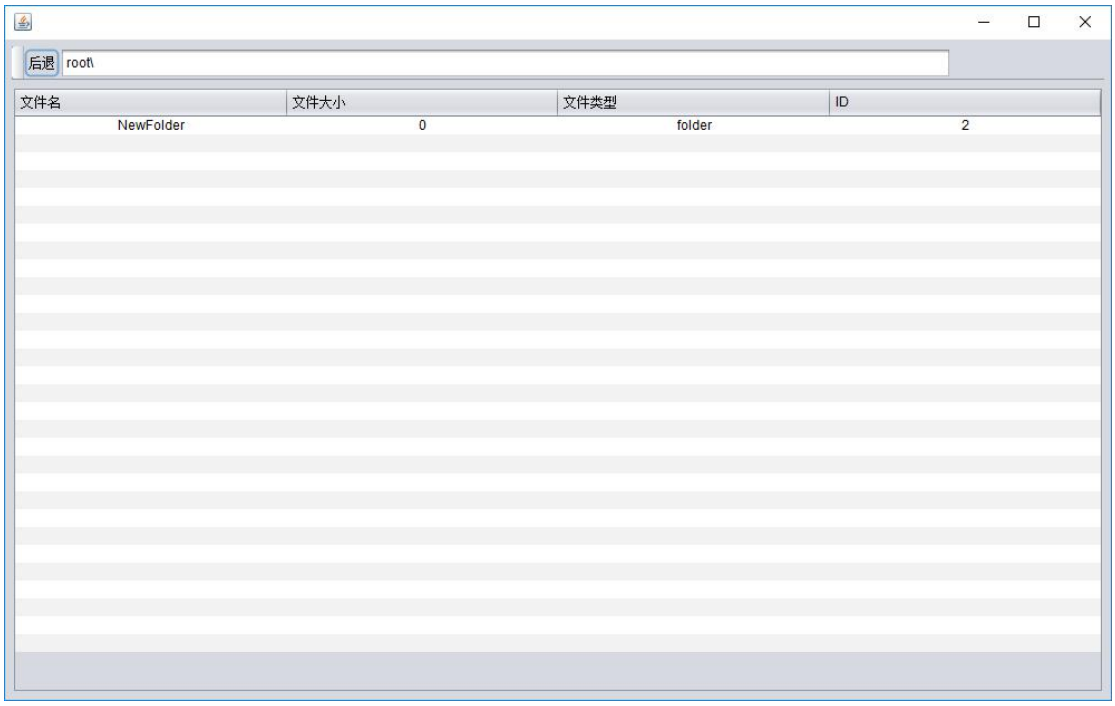
在表格上右键可以弹出菜单，如下图：



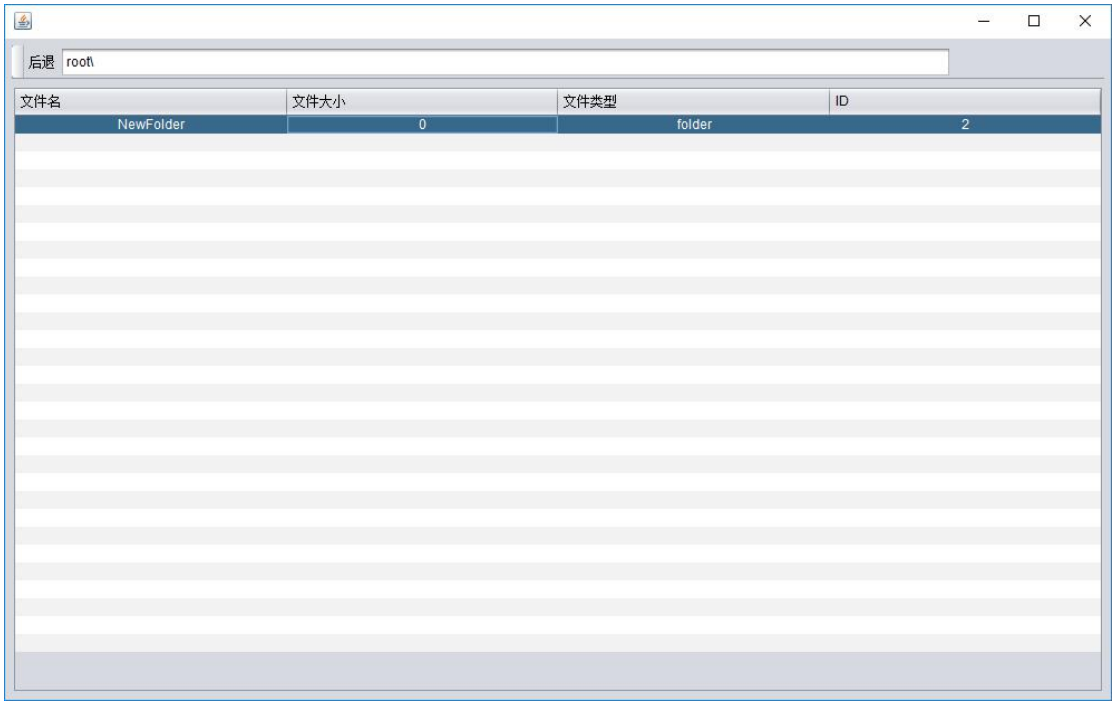
菜单中一共有三项。第一项是新建一个文件夹，可以点击该文件夹创建一个新文件夹，并进入这个文件夹，如下图：

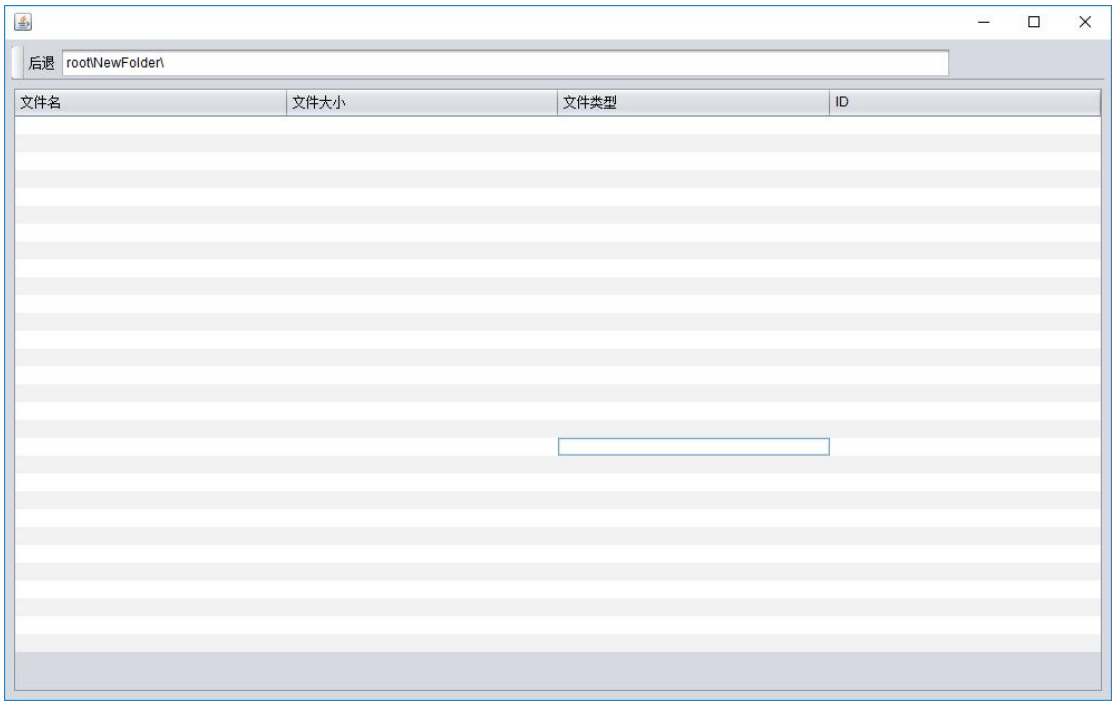


下图为新建了文件夹：



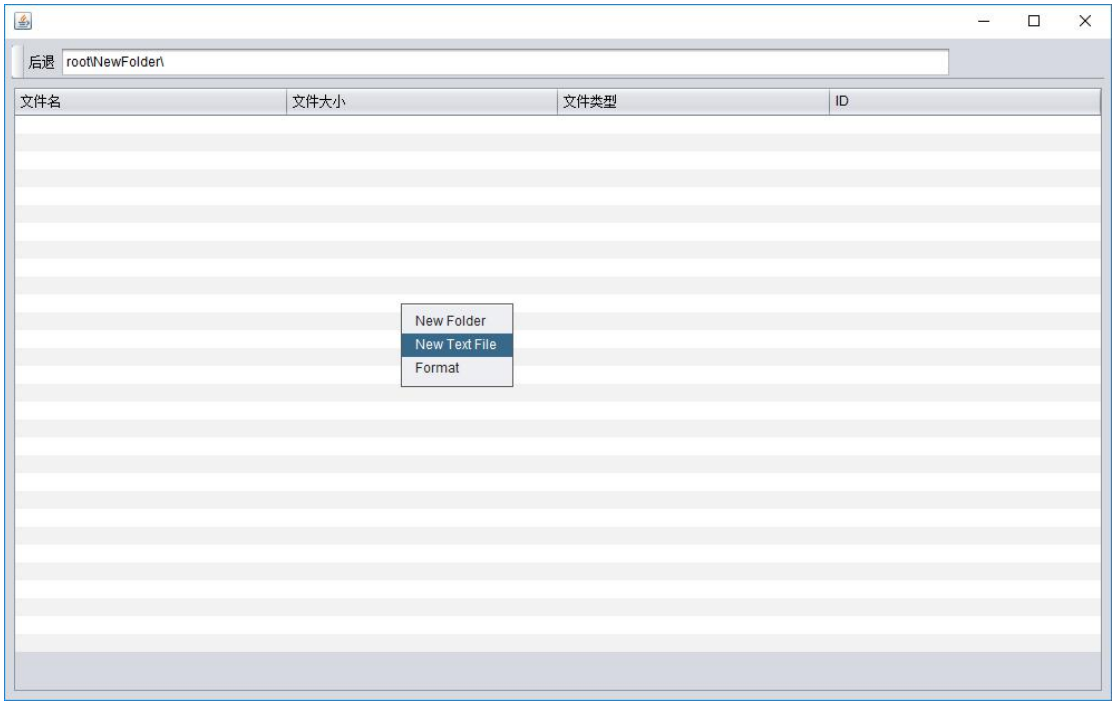
左键双击该文件夹可以进入该文件夹：

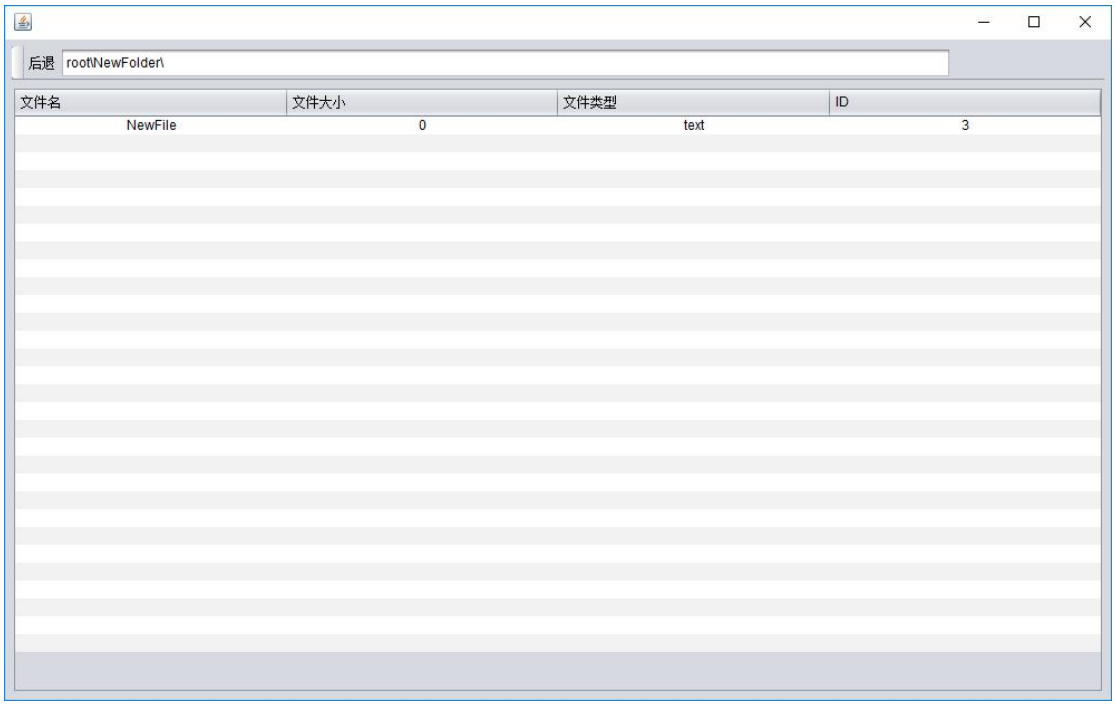




观察上图中程序的路径，已经进入了该文件夹。

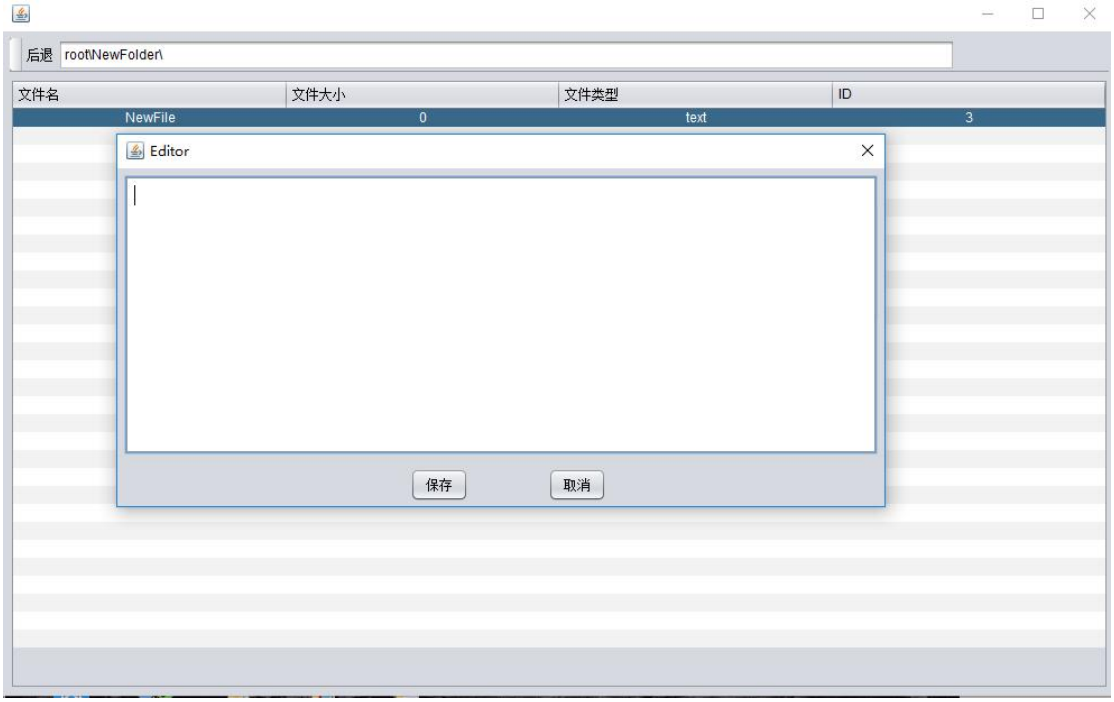
在该文件夹内再右键单击，选择第二个选项可以新建一个文本文档：



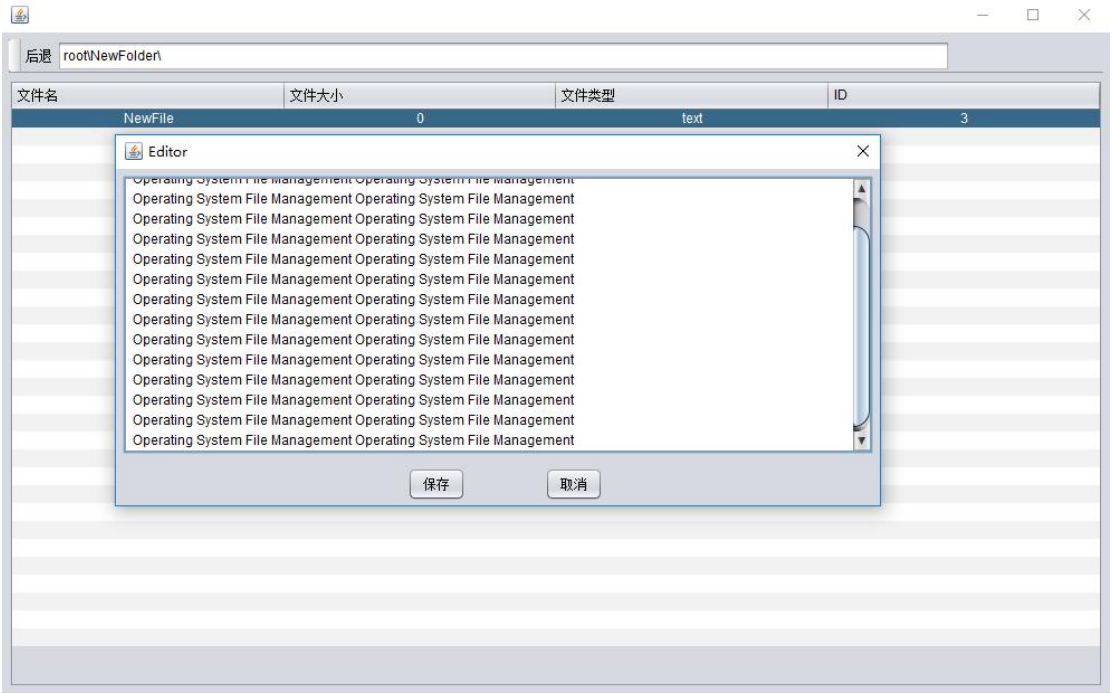


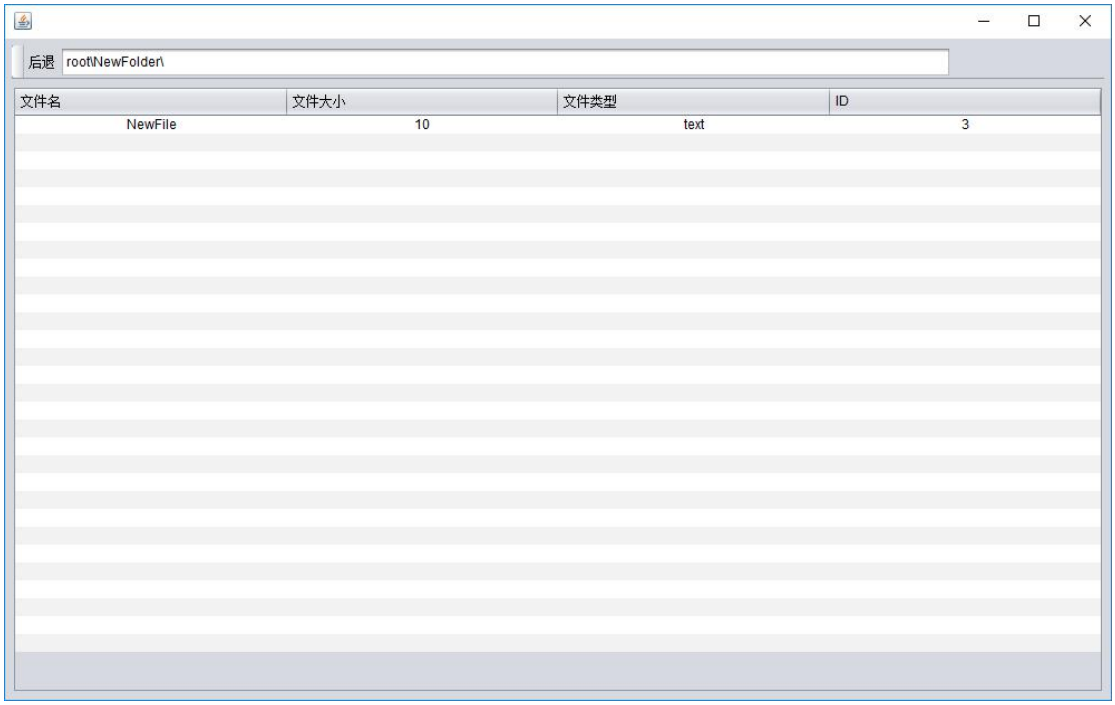
观察上图，已经在该文件夹内新建了一个文件。但是该文件的大小为 0，因为还没有向文件中写入任何内容。

双击新建的文件，可以打开一个文本编辑窗口，向文件内写入内容。如下图：

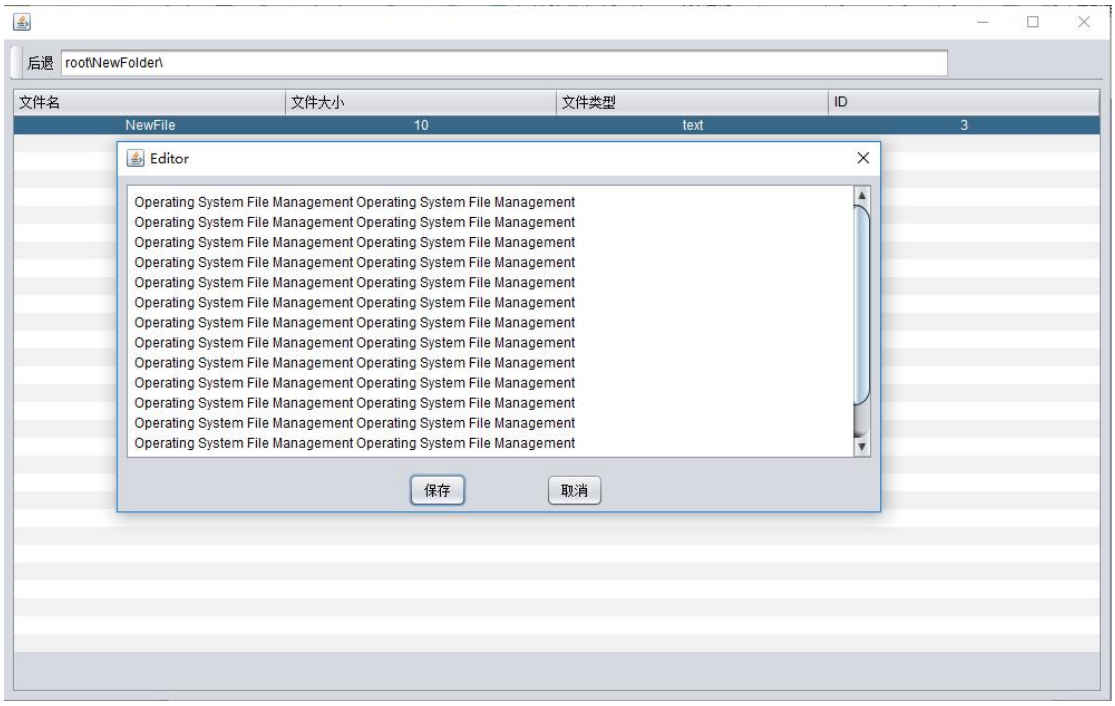


向文本框内输入内容后按确定，可以发现文件大小已经变化。再次打开该文件，发现内容还在。



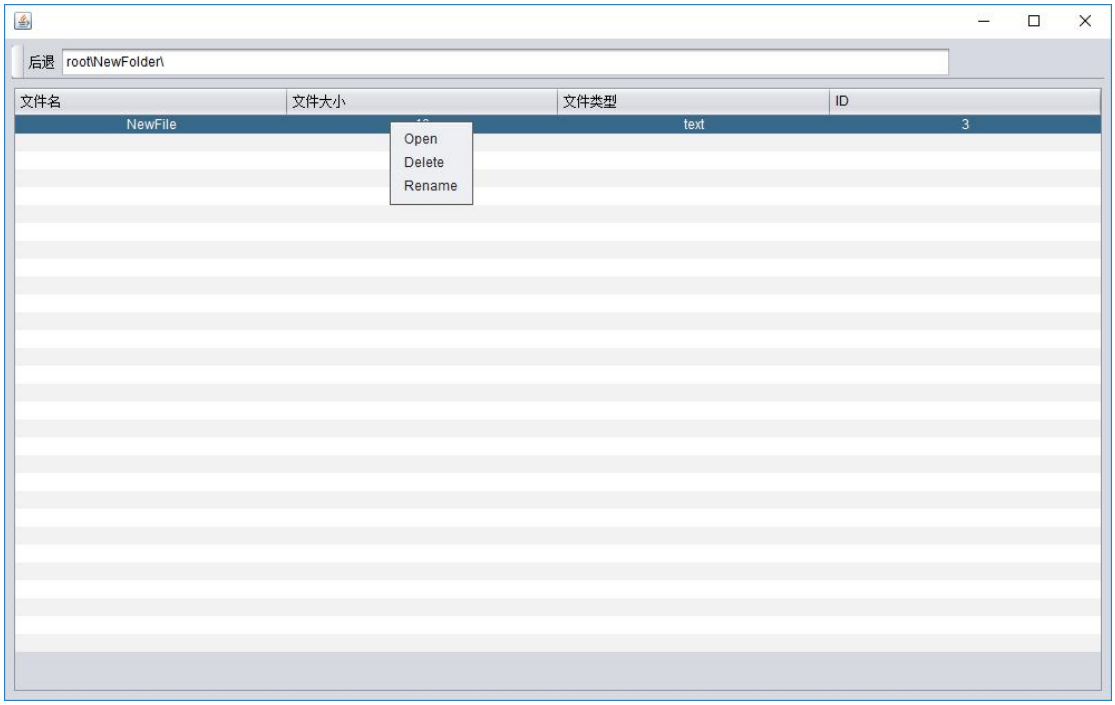


再次打开该文件，可以看到文件内容，此时文件大小是 10：

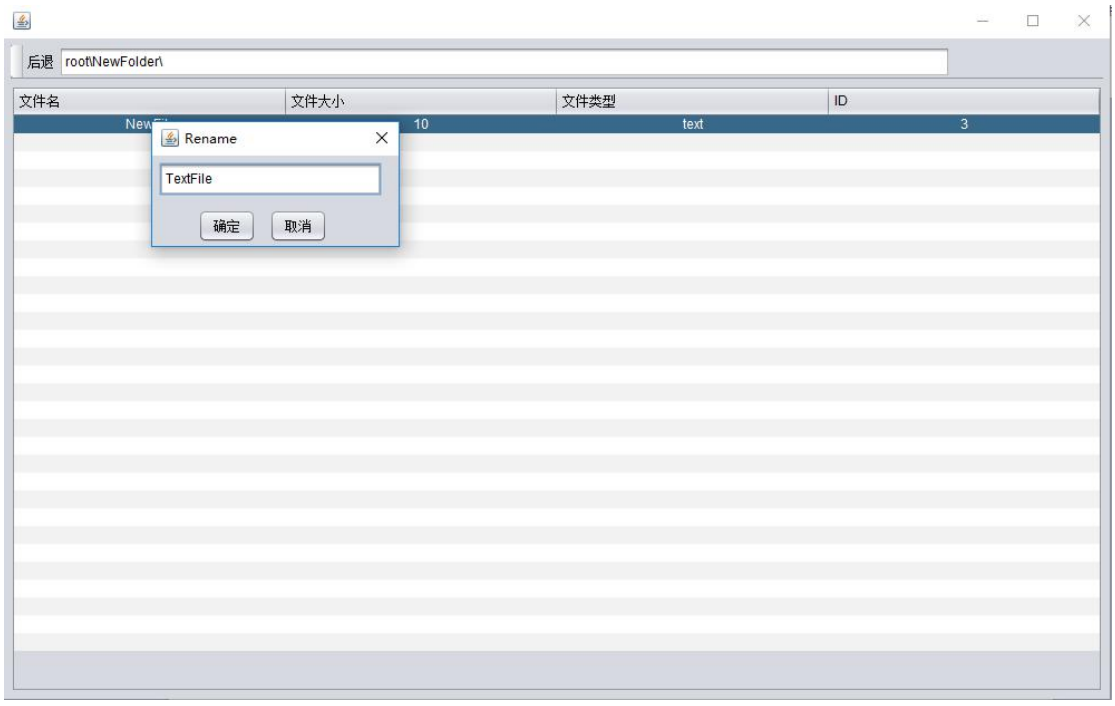


还可以通过右键菜单对文件进行打开，重命名和删除操作。选定文件（对应文件栏为蓝色），

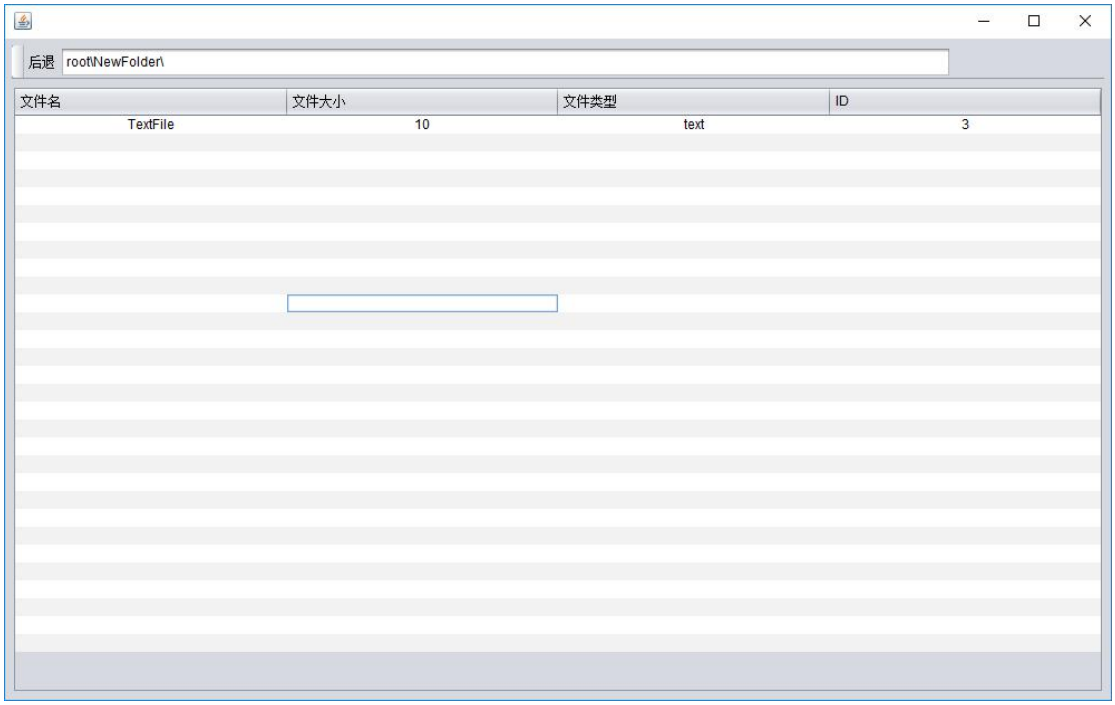
之后单击右键，会弹出右键菜单，如图所示：



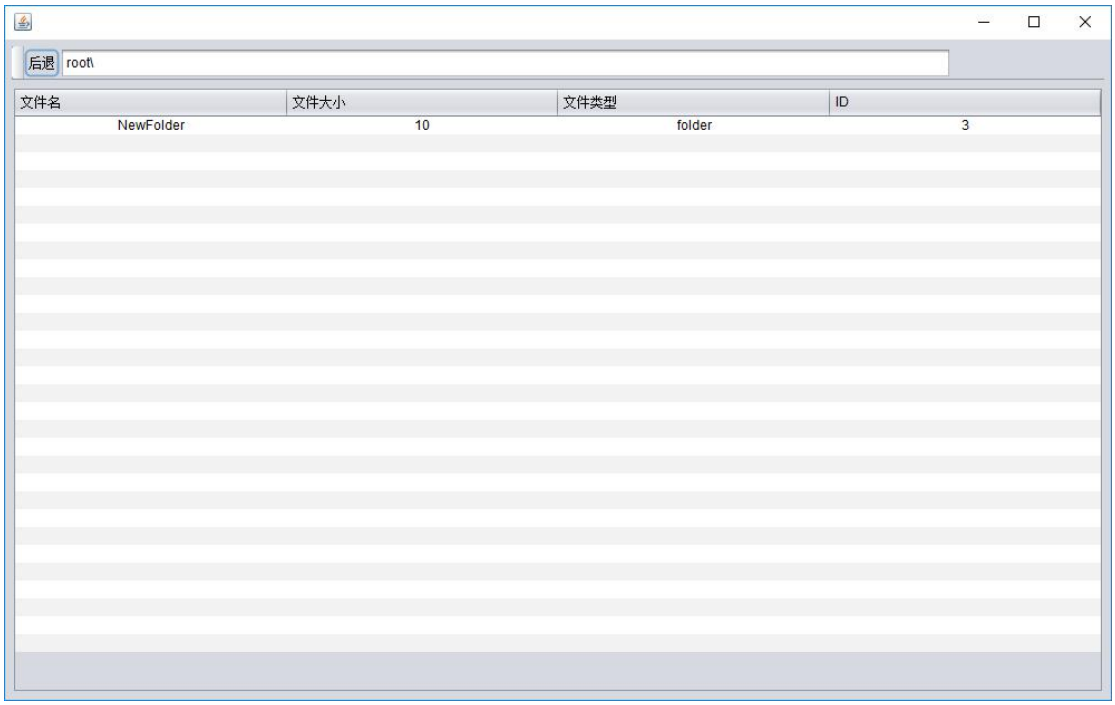
点击 Rename 后，可以对文件重命名，如下图所示：



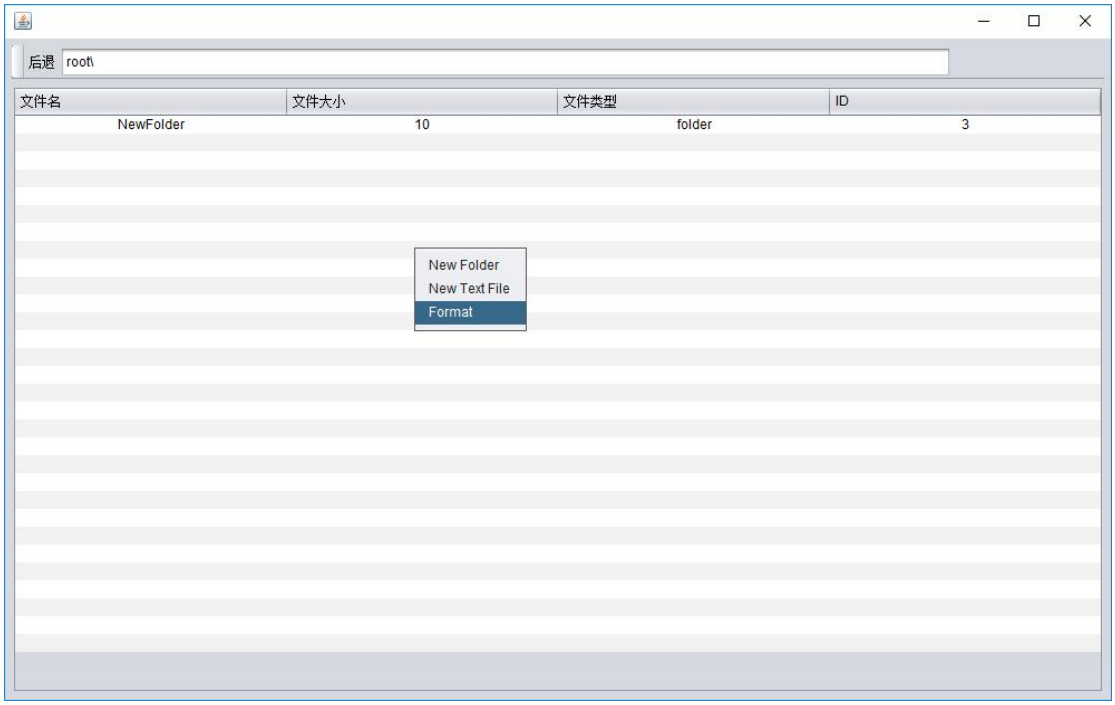
点击确定后观察到文件名已经改变了，如下图：

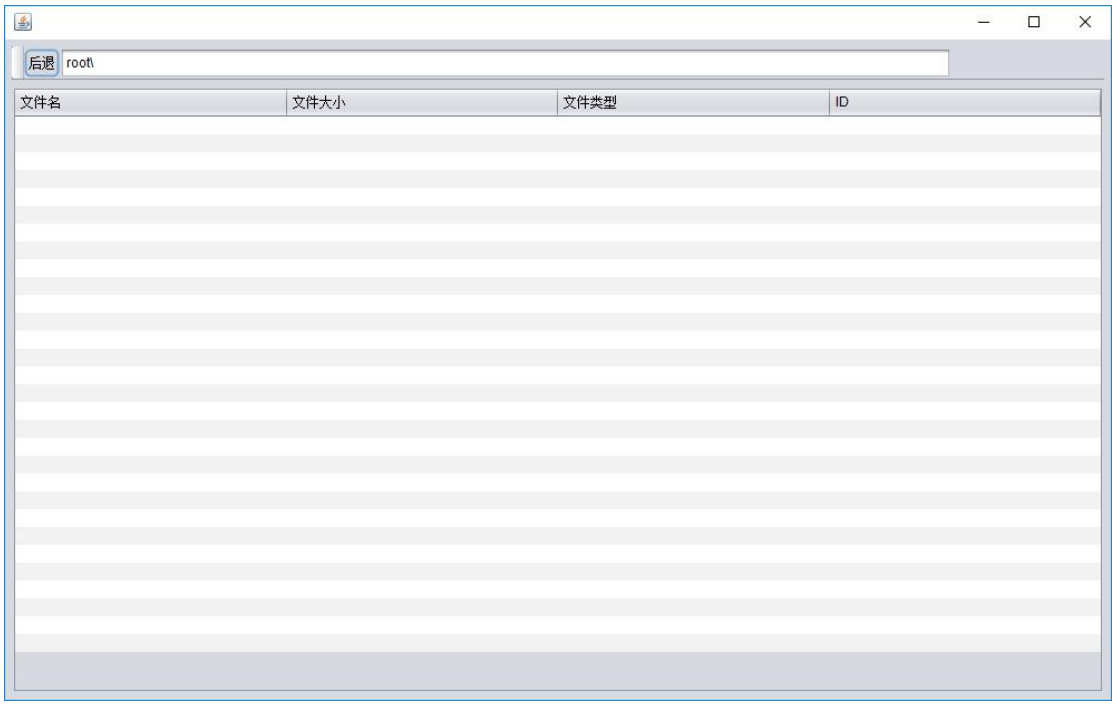


点击后退，可以回到根目录，注意到此时文件夹的大小为 10。



在空白处右键弹出右键菜单后，选择第三项，可以格式化当前目录，如下图：





四、实现方法：

程序代码一共由四个类组成。

FCB 类定义了每个文件的文件控制块，保存了文件的名称，ID，文件大小，子文件的 ID，

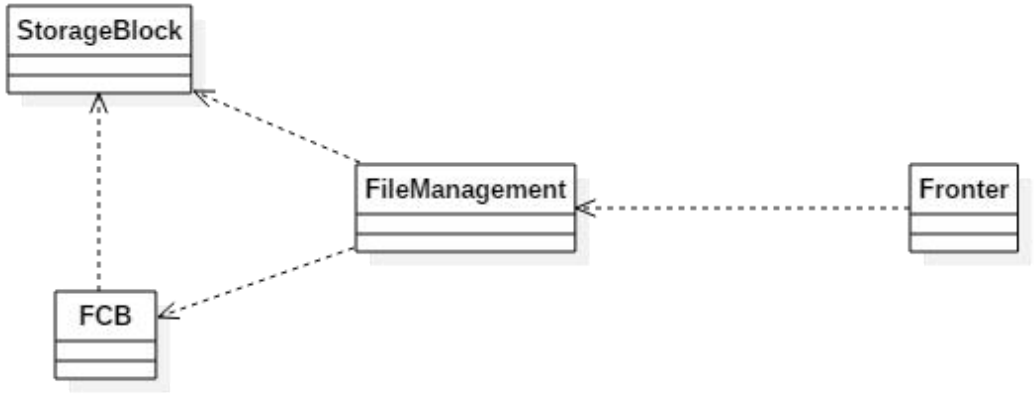
文件类型，文件父级和占用的文件存储块的列表。同时，也提供了一些文件的基本操作，比如写入，读取，获取子文件列表，获取文件存储块列表，获取文件类型，获取文件大小和一些设置文件属性的函数。

StorageBlock 类定义了文件存储块，保存了文件存储块的 ID，状态（被占用或没有被占用），文件存储块所属的文件以及文件存储块所保存的内容。同时也提供了对文件存储块的基本操作，比如写入，读取，获取文件存储块的属性，修改文件存储块的属性。

FileManagement 类是一个单例类，负责文件管理系统的调度。在这个类中保存了两张全局的表。一张是所有的 FCB 对象的表，另一张是所有文件存储块的表。并且提供了根据 ID 获取 FCB 对象和文件存储块对象的方法。同时还定义了 load 函数和 store 函数。load 函数在程序启动时从硬盘上读取文件管理系统的信息。store 函数在程序关闭时把文件管理系统中更新的信息写回硬盘，以便于下次打开系统时可以还原。FileManagement 类还记录当前所在的文件对象。同时，也提供了向特定文件写入内容和从特定文件读出内容的方法。这些方法都是以 FCB 的 ID 为输入和检索的依据，并通过该类中根据文件 ID 获取 FCB 对象的函数来得到 FCB 对象。另外，FileManagement 类还用位图记录了各个文件存储块的空闲情况。

Fronter 类是一个界面类。也是程序的入口所在。它初始化了整个界面，并通过调用 FileManagement 提供的方法来对文件系统进行修改，同时也会根据文件结构实时对界面上的信息进行更新。在 Fronter 类中还通过对鼠标的监听来弹出菜单，并通过匿名内部类的方式实现了对鼠标事件的响应。

三个类之间的关系用 UML 表示可为：



五、代码实现：

FCB：

权限	返回值	函数名	参数	说明
public	int	getFileID		获取文件的 ID
public	String	getFileName		获取文件名
public	void	setFileName	String filename	设置文件名
public	int	getFileType		获取文件类型
public	void	setFileType	int type	设置文件类型
public	int	getFileSize		获取文件大小
public	int	getParent		获取父文件的 ID
public	ArrayList<Integet>	getStorageBlockTable		获取占用的文件存储块的表
public	ArrayList<Integer>	getSubFiles		获取子文件的 ID 的表
public	boolean	addSubFile	FCB subfile	添加子文件

public	void	addSize	int size	增加文件的大小
public	void	setSize	int size	设置文件
public	void	clear		清除该文件的子文件

StorageManagement:

权限	返回值	函数名	参数	说明
public	boolean	getState		获取文件存储块的状态
public	void	changeState		改变文件存储块的状态
public	int	getIndex		获取文件存储块的 ID
public	booean	setIndex	int index	设置文件存储块的 ID
public	FCB	getFileObj		获取所属的文件的 FCB
public	void	setFileObj	FCB file	设置所属文件
public	String	write	String text	写入内容，返回剩下的内容
public	String	read		读出内容

FileManagement：

权限	返回值	函数名	参数	说明
public	void	format		格式化当前目录下的文件
public	int	getFCBSize		获取文件总数
public	void	createFolder	String name, int fileid	新建一个文件夹
public	void	createFile	String name, int fileid	新建一个文本文件
public	void	deleteFile	int fileid	删除文件
public	String	readFile	int fileid	读取文件
public	int	getFreeBlock		获取一个空闲文件存储块
public	boolean	writeFile	FCB file, String text	写文件
public	void	load		从硬盘加载文件
public	void	store		把文件信息存储到硬盘上
public	FCB	getFCB	Integer ID	通过 ID 获取 FCB 对象
public	FCB	getCurrentFolder		获取当前目录
public	void	setCurrentFolder	FCB fcb	设置当前目录
private	void	load_fcb		加载 FCB 信息