# 操作系统第三次作业

——1753603 林寅嘉

目录

[操作系统第三次作业 1](#_Toc10885835)

[1 界面实现 1](#_Toc10885836)

[1.1 主界面 1](#_Toc10885837)

[1.2 文件属性面板 2](#_Toc10885838)

[1.3 警告窗口 3](#_Toc10885839)

[1.4 txt文件界面 4](#_Toc10885840)

[2 各项功能的实现 5](#_Toc10885841)

[2.1 格式化 5](#_Toc10885842)

[2.2 创建子目录 6](#_Toc10885843)

[2.3 删除子目录 7](#_Toc10885844)

[2.4 显示目录 8](#_Toc10885845)

[2.5 创建文件 9](#_Toc10885846)

[2.6 打开文件 9](#_Toc10885847)

[2.7 关闭文件 10](#_Toc10885848)

[2.8 写文件 11](#_Toc10885849)

[2.9 读文件 11](#_Toc10885850)

[2.10 删除文件 12](#_Toc10885851)

[3 保存以及读取信息 12](#_Toc10885852)

[4 具体技术细节 13](#_Toc10885853)

[4.1 文件储存管理 13](#_Toc10885854)

[4.2 空闲空间管理 13](#_Toc10885855)

[4.3 文件目录 13](#_Toc10885856)

[5注意事项 14](#_Toc10885857)

## 1 界面实现

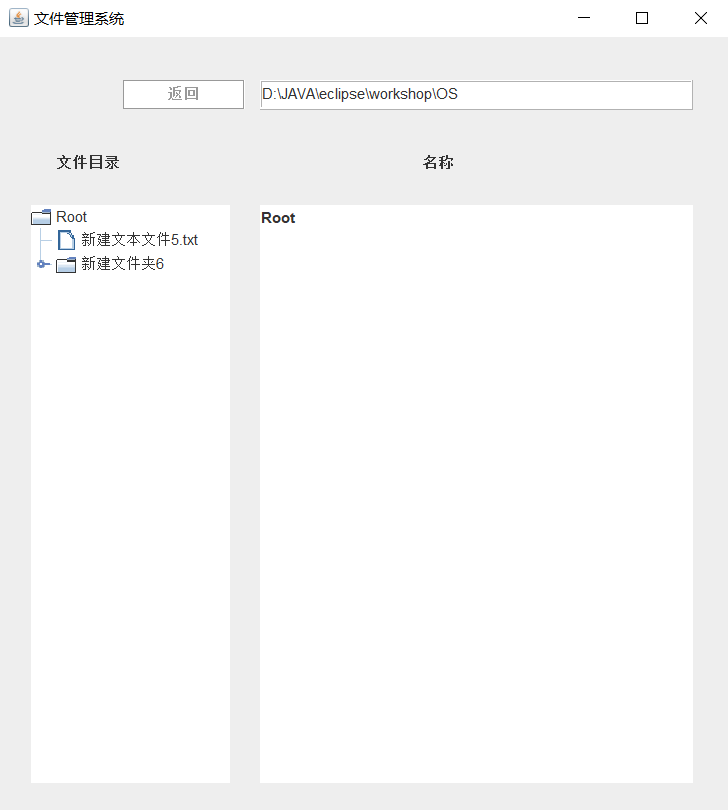
共实现了四个界面：文件管理主界面，文件属性界面，对文件操作失败后的警告界面以及打开txt文件时的界面。

## 主界面

使用WindowBuilder设计，参考了Windows的界面，界面截图如下：

首先，上面可以显示当前所在的位置的绝对路径，以及一个返回按钮，用于返回上一级目录（当处于根目录时无法返回，注意本程序默认的根目录时Root），下面左侧有当前文件系统的树形结构，用户可以通过其清除的看到文件的结构，右下方为当前所在目录中包括的文件（共有两种类型的文件，文件夹和txt文件，用户可以通过后缀来区分他们）。

详细代码请参考MainWindow.java

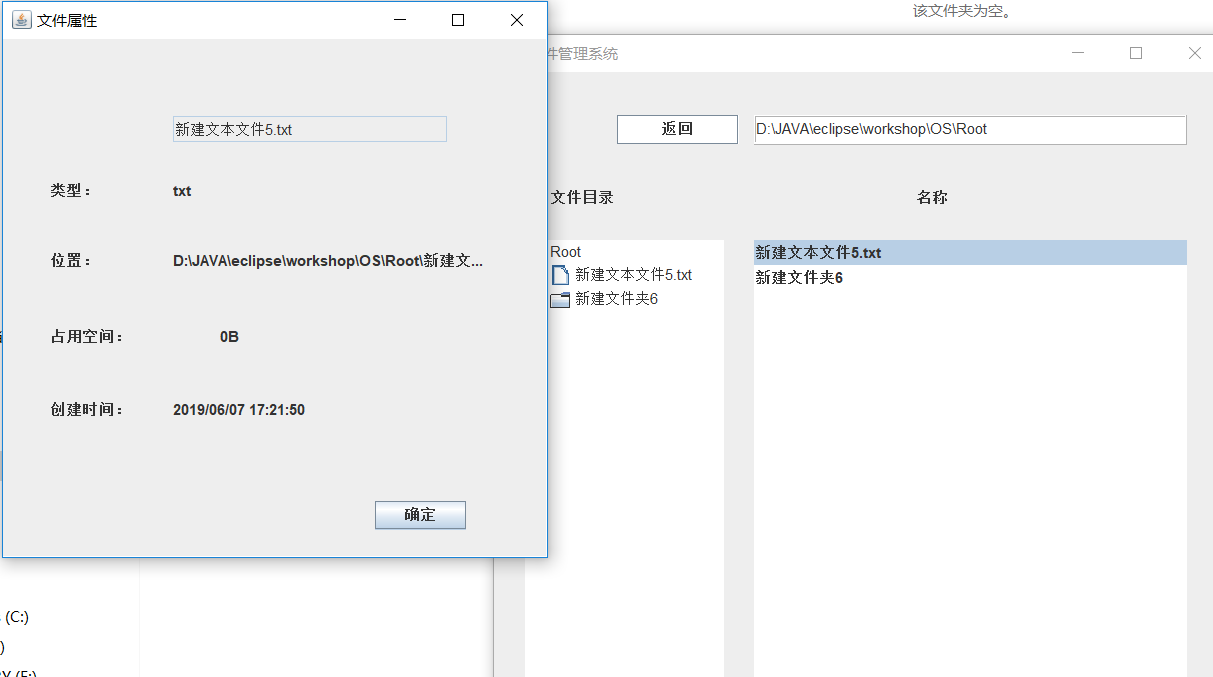


## 文件属性面板

属性面板包括了文件的名称，类型，绝对路径，大小以及创建时间，截图如下：

用户可以通过右键文件夹或者txt文件来获取所选文件的属性。

详细代码请参考FileAttrbutes.java



## 警告窗口

考虑到用户的一些误操作以及内存分配错误的情况，本程序提供了错误警告窗口。截图如下：

警告分为以下5种：通过switch-case语句来判断，代码如下：

详细代码请参考AlertWindow.java



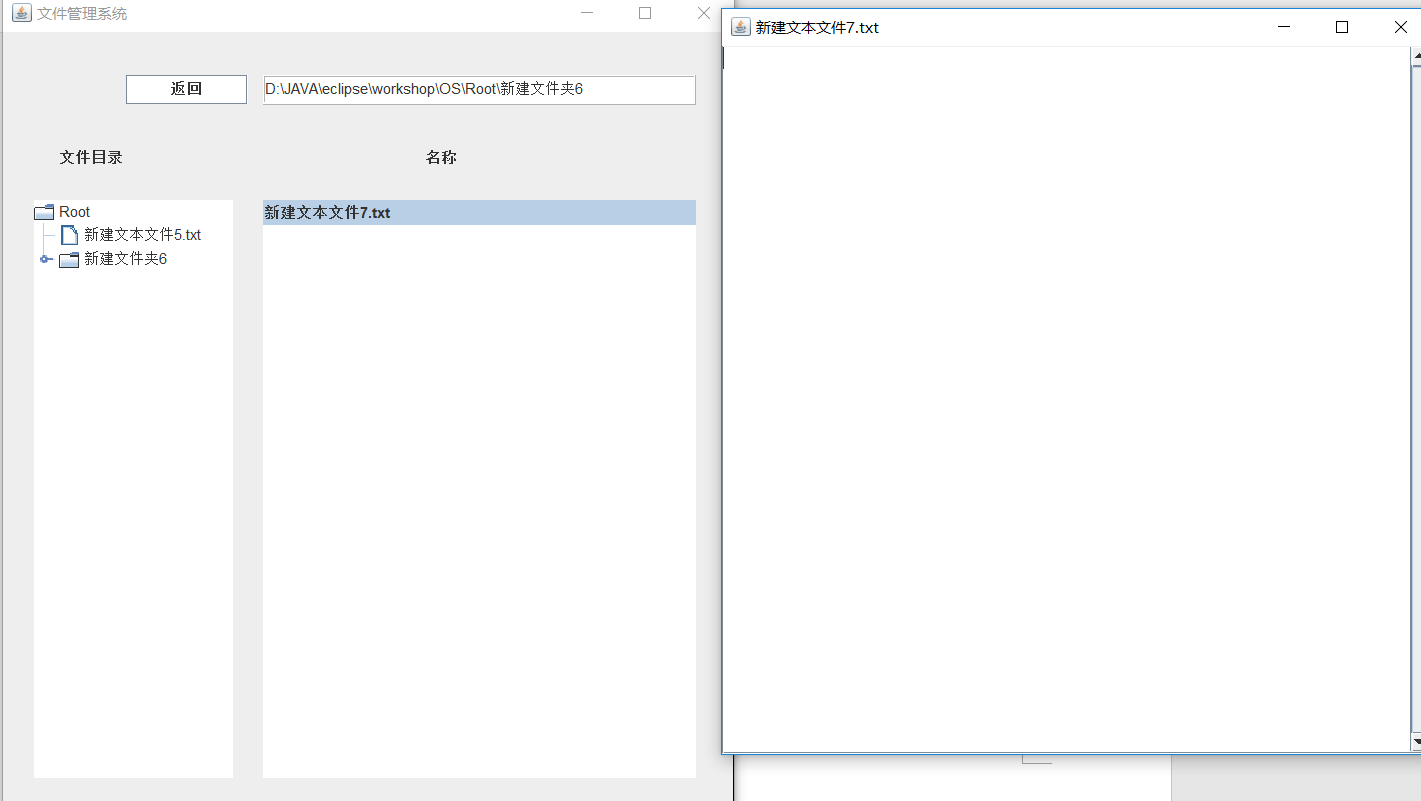


## txt文件界面

当用户双击txt或者右键->打开时，会弹出类似于文本编辑的窗口，此时用户可以对其进行删除或写入。

详细代码请参考NotePad.java

界面如下：

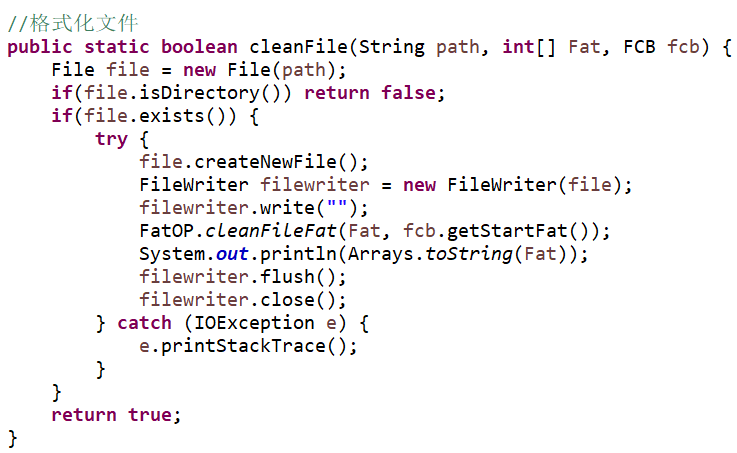
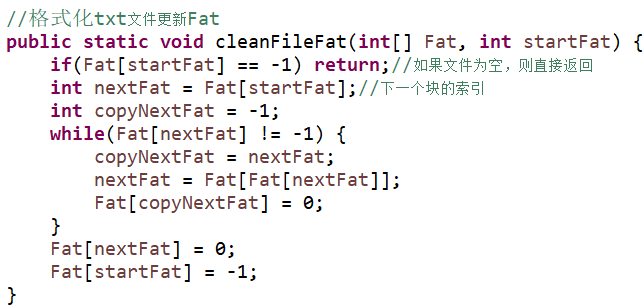


## 各项功能的实现

## 格式化

本程序只支持txt文件的格式化，格式化时，先要将磁盘上的文件内容清空，再将文件所对应的Fat表中的对应值变为0。

对应的方法为FatOP.cleanFileFat()以及FileOP.cleanFile(),代码如下：



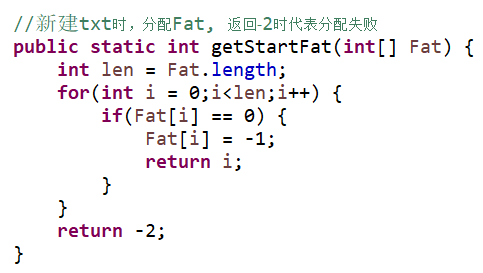
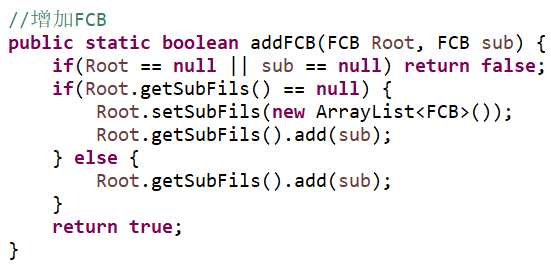
## 创建子目录

用户可通过右键来选择要创建的文件类型，创建文件时要给文件分配相应的Fat，以及创建对应的Fcb。

对应的方法为FileOP.makeRoot, FileOP.makeDir,FileOP.makeFile,

FatOP.getStartFat, FcbOP.addFCB

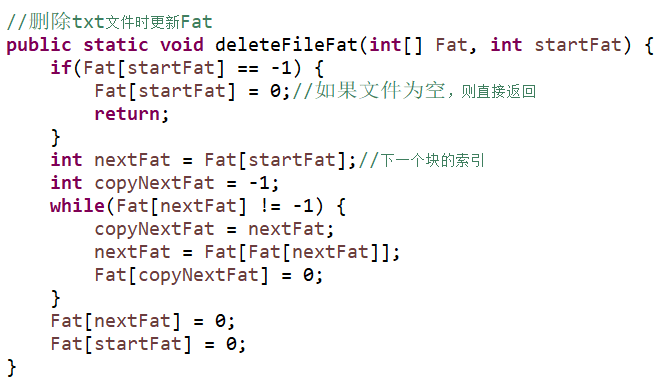
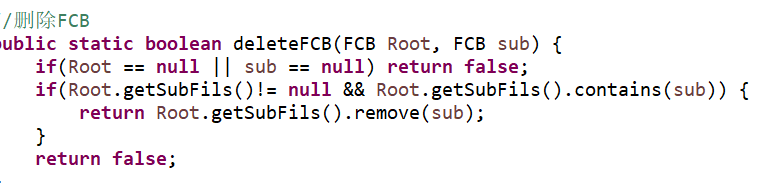
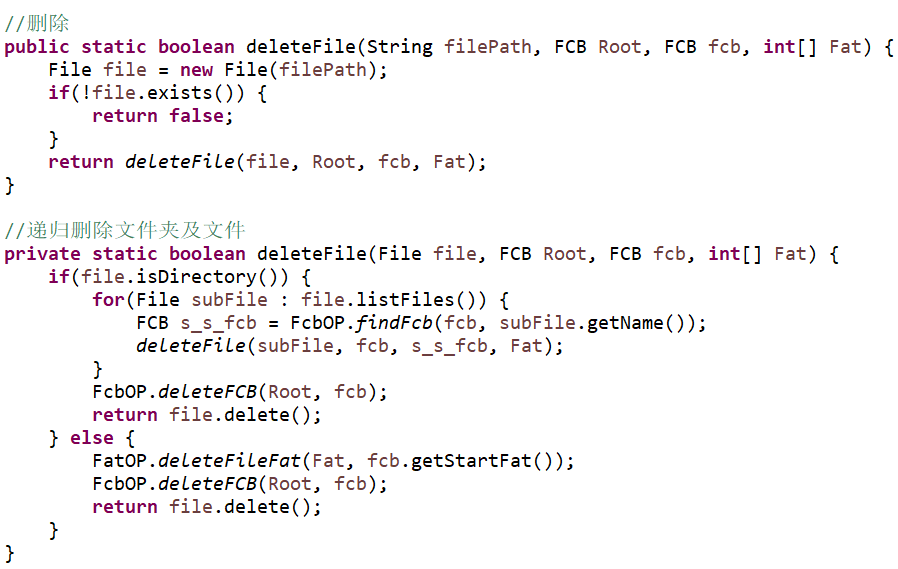
部分代码如下：



## 2.3 删除子目录

用户可通过右键->删除来删除对应的文件，考虑到文件夹种可能会包含文件，所以需要递归的调用函数将文件夹删除，删除时要删除文件对应的Fat以及Fcb，同时还要更新UI(JTree， JList)

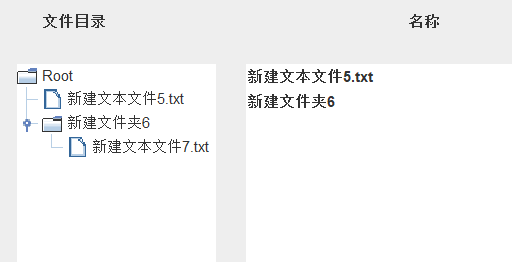
部分代码如下：



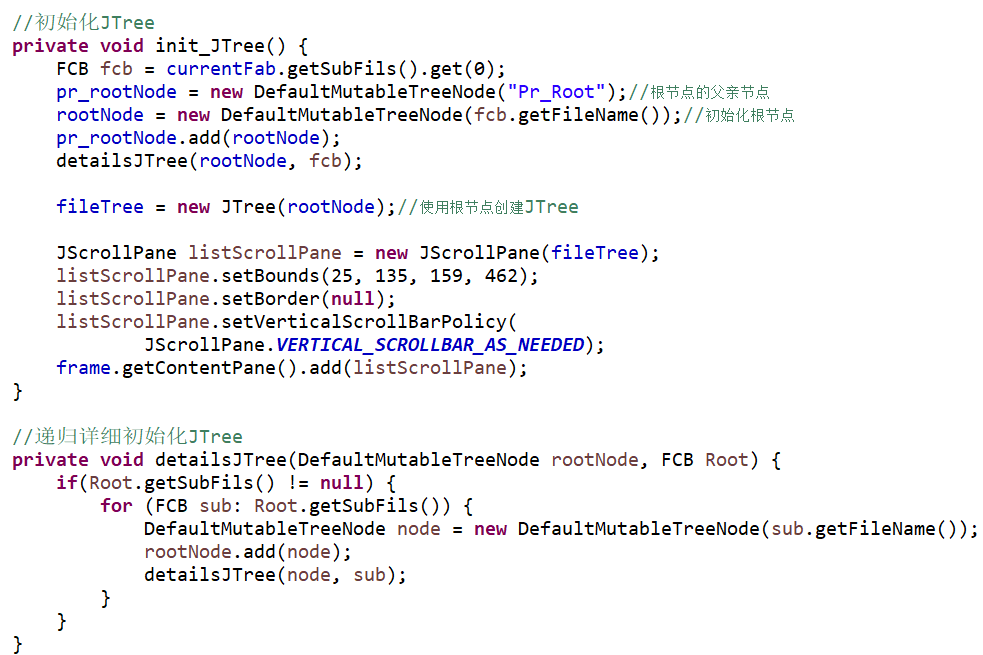
## 显示目录

本程序提供树状结构来显示当前的文件结构，通过JTree来实现。

截图如下：



代码如下：



## 创建文件

详情请看2.2 创建子目录

## 打开文件

本程序提供两种打开文件的方式，鼠标右键或者双击文件。如果打开的是文件夹则显示下级目录，若打开的时txt，则弹出txt编辑窗口。

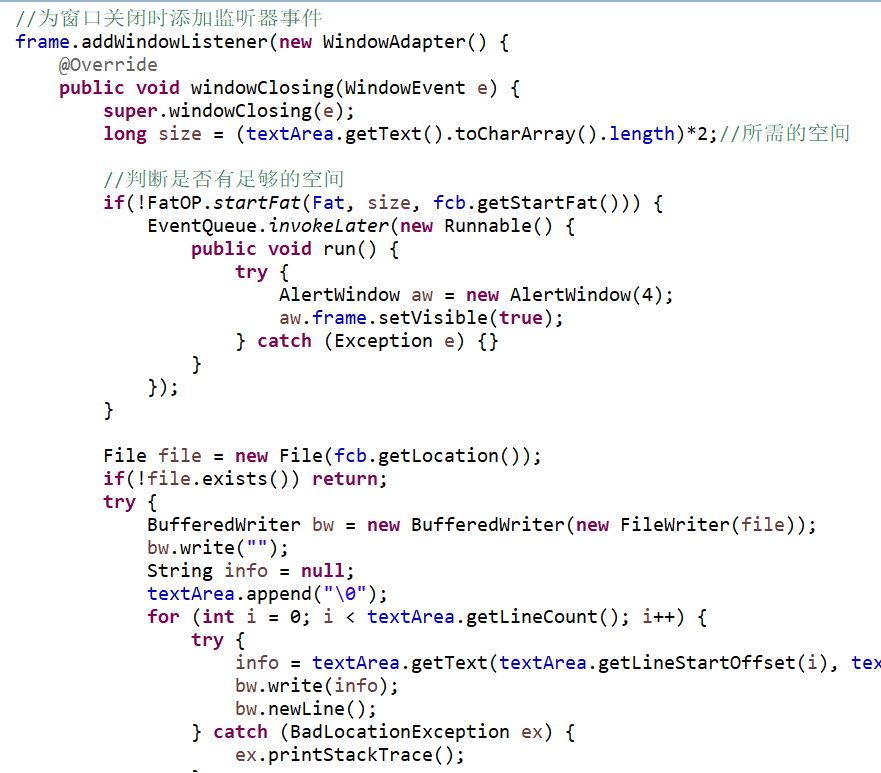
详情请见MainWindow.RightMouseEvent(), MainWindow.init\_JPopupMenu(),

部分代码如下：



## 关闭文件

点击文本编辑器的关闭按钮即可，在关闭的同时需要将文件的内容写入到磁盘对应的文件中，并修改文件的Fat，size信息。代码如下：

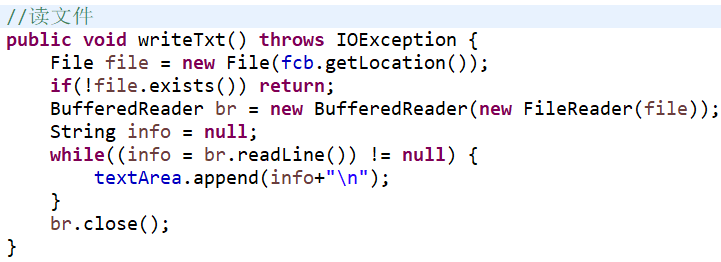


## 写文件

打开txt编辑器，在文本框中写入即可。详情请见2.7

## 读文件

打开文件时，需将对应磁盘中文件的内容写入到txt编辑器中，代码如下



## 删除文件

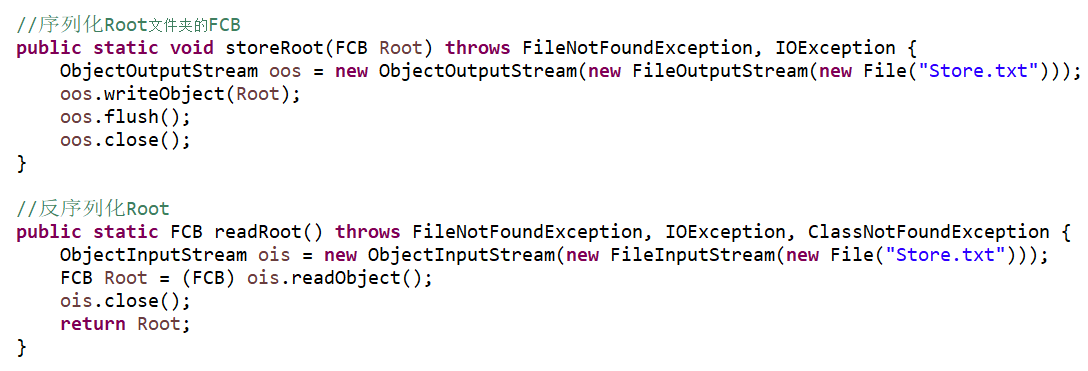
详情请见2.3

## 保存以及读取信息

在关闭程序时需将文件对应的FCB, FAT信息写入到磁盘中，以便下次打开程序时还原上一次的信息。所以使用了序列化以及反序列化来写入，读取对象信息。

详情请见FcbOP.storeRoot, FcbOP.readRoot, FatOP.writeFat, FatOP.readFat,

部分代码如下：



## 具体技术细节

## 4.1 文件储存管理

文件存储管理使用Fat表的方式，利用一个长度为64的数组来代表磁盘上对应块的存储情况，0代表对应的块未被利用，-1代表文件结束，每一个单元格中都存储着文件下一个占用块的索引，形成链接结构。

## 4.2 空闲空间管理

空闲空间管理采用了将位图与Fat表合二为一的方案。

## 4.3 文件目录

文件目录采取了多级目录结构，详情请见FCB.java，每一个FCB中都存储了父级目录的指针，以及子集目录的指针，代码如下：



## 5注意事项

第一次运行程序并关闭程序后，会在exe文件所在目录下生成Root文件夹，Store.txt(保存文件FCB信息)，Fat.txt（保存Fat），请勿随意删除其中一个。如不小心删除，请删除其他文件，重新运行程序即可（但是原先所有的文件信息将会被清除）