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： 2020 学年第一学期

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 小组成员、组内分工、工作量比例、各成员个人成绩 | | | | | | | | | | | |
| 学 号 | 2018xxx  xxx | 姓 名 | xxx | 分 工 |  | | | 工作量  比例 |  | 成绩 |  |
| 学 号 | 2018xxx  xxx | 姓 名 | xxx | 分 工 |  | | | 工作量  比例 |  | 成绩 |  |
| 学 号 | 2018xxx  xxx | 姓 名 | xxx | 分 工 |  | | | 工作量  比例 |  | 成绩 |  |
| 学 号 | 2018xxx  xxx | 姓 名 | xxx | 分 工 |  | | | 工作量  比例 |  | 成绩 |  |
| 实 验 题 目 | 模拟磁盘文件系统实现 | | | | | | | | | | |
| 自 我 评 价 |  | | | | | | | | | | |
| 教 师 评 语 |  | | | | | | | | | | |
|  |  | | | | | 教师签名 | 孙微微 | | | | |

一、 需求分析

设计一个简单的文件系统，用文件模拟磁盘，用数组模拟缓冲区，要求：

(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

1. 采用一个文件来模拟电脑磁盘。

2. 本程序共个 12 模块，分别如下：

(1) 主程序模块 HelloController()

(2) 界面初始化模块 mainview()

(3) 模拟文件模块 file()

(4) 模拟文件夹模块 folder()

(5) 模拟磁盘块模块 block()

(6) 文件分配表模块 fat()

(7) 已打开文件模块 fileopened()

(8) 文件创建模块 createfile()

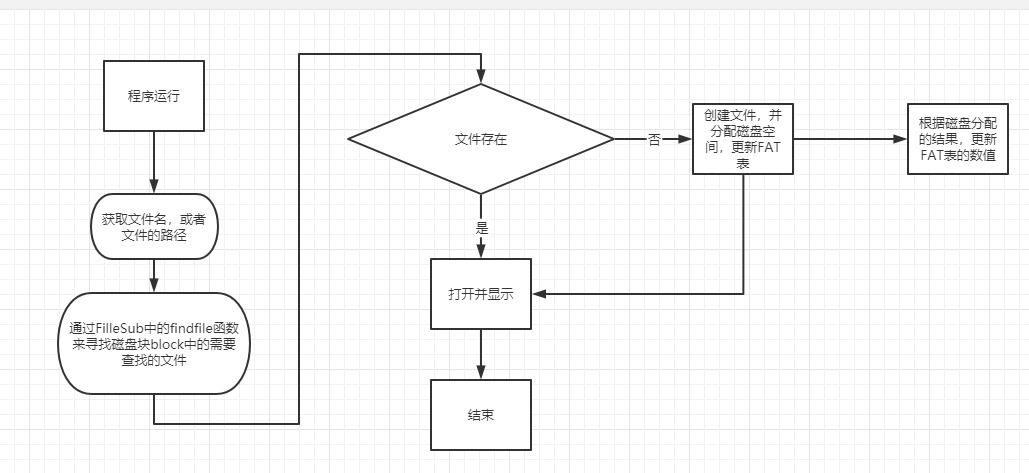
(9) 文件夹创建模块 createfolder()

(10) 命令行删除文件(夹)操作模块 delete\_pathfile()

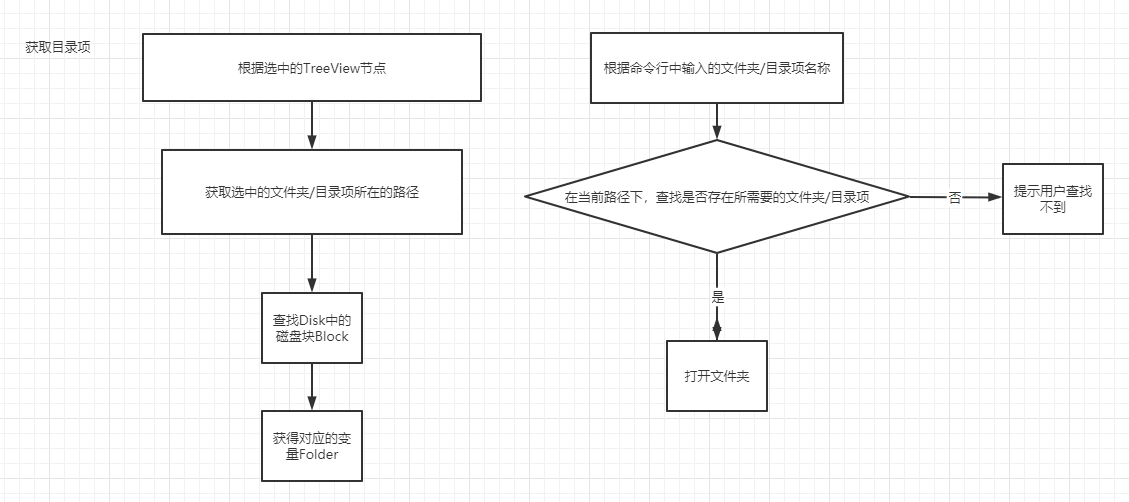
(11) 文件读写或只读打开模块 openfile()

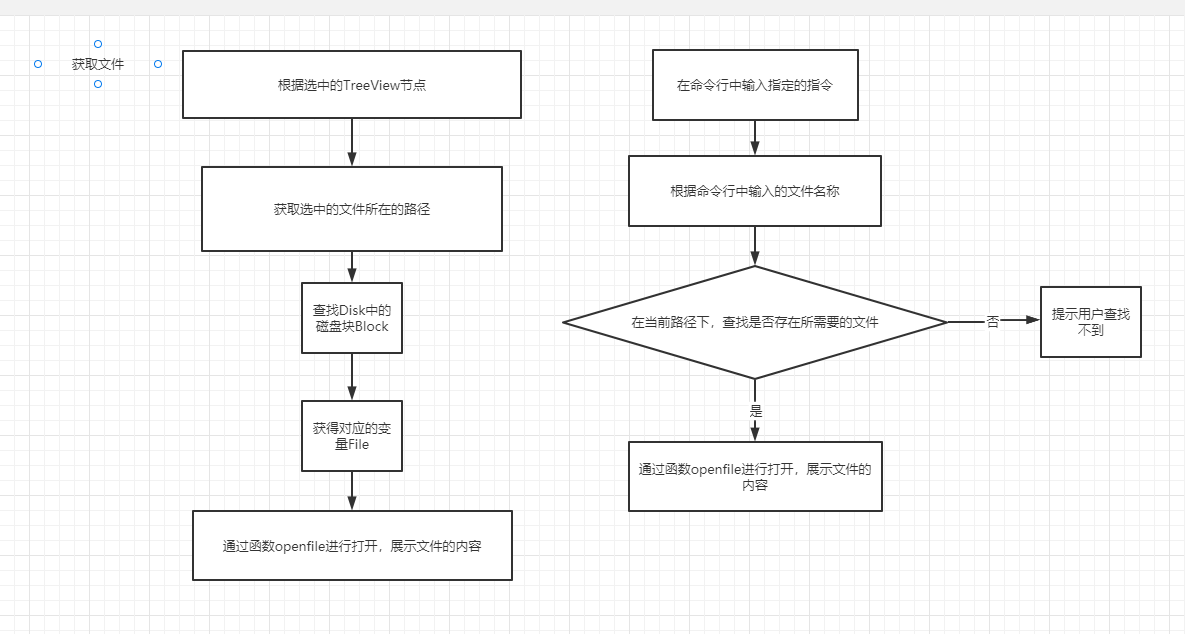
(12) 文件夹打开模块 openfolder()

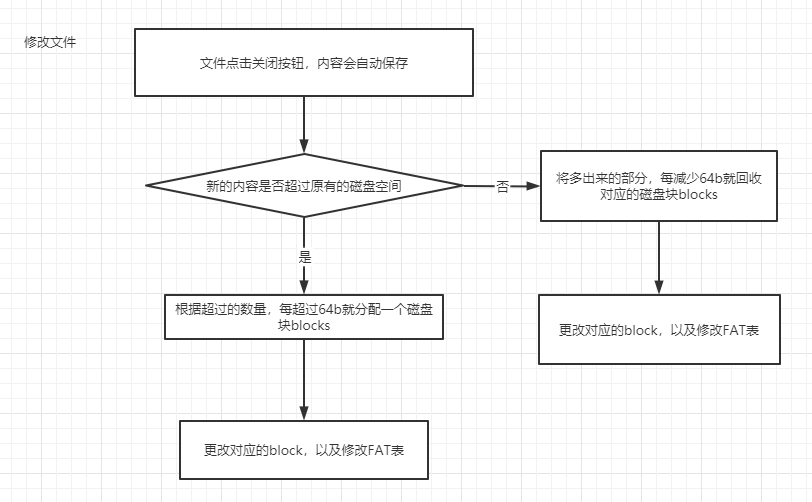
3. 主程序流程图



4. 各程序模块之间的层次关系





三、 详细设计

1. 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， 254 代表该块损坏无法使用。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。

读写方法：使用 64B 的 byte 数组 readBuffer 和 writeBuffer,读取或写入时先将文件指针移动到对应位置，然后以块为单位进行存取。

1. 模拟文件模块File()

文件类File {  
 String fileName;//文件名  
 int flag;//只读/读写标志  
 int num;//记录文件的起始盘块号  
 String content;//文档中保存的字符串  
 int size;//占用的字节大小  
 Folder parent;//父母文件夹  
 String path;//路径S  
 boolean open;//记录是否打开  
 int type;//记录文件的属性  
 public File(String fileName,String path,int num,Folder parent,int type,int flag)//新建文件，依次输入名字，所在路径，文件的起始盘块号，父母文件夹，文件的属性，只读/读写标志  
 {  
 依次将File类的数据初始化  
 disk D=FileSub.Disk;  
 D.blocks[num].BlockChange(-1,this,true);

更新磁盘上的内容  
 }  
 功能函数changeFileContent(String content) throws UnsupportedEncodingException//更新文件中的内容content  
 {  
 this.content=content;//更改对应的字符串  
 if(新的内容所占用的磁盘位置变少)//需要回收磁盘  
 {

更新磁盘block

同时更新FAT表，将多占用的地方归还  
 }  
 else if(新的内容所占用的磁盘位置变多){

更新磁盘block

同时更新FAT表，找到FAT中空闲区域，并修改  
 }  
 更改size的大小  
 }

1. 模拟文件夹模块 Folder()

文件夹类Folder（）{

String folderName;//文件夹名字  
int num;//起始盘块号  
int size=64;//占用的字节大小，目录都是64的字节大小，占据八个目录项  
Folder parent;//父母文件夹  
String path;//路径  
ArrayList<Object>children;//孩子节点  
int type;//记录目录属性  
ArrayList<String>item=new ArrayList<>();//用字符来直接记录目录项，方便观察  
  
public Folder(String folderName,String path,int num,Folder parent)  
{  
 对相应的数据进行赋值

初始化所有数据  
}  
public boolean addChildrenNode(Object object)//用于添加孩子节点的函数  
{  
 if(该节点的目录项已经超过8个)  
 return false;  
 if(如果新创建的子节点属于文件类)  
 {  
 children.add(object);//向孩子节点中添加对应的子节点  
 更新item中的数据  
 }  
 if(如果新创建的子节点属于文件夹类)  
 {  
 children.add(object);//向孩子节点中添加对应的子节点  
 更新item中的数据  
 }  
 return true;  
}

}

1. 模拟磁盘块模块block()

磁块类Block {  
 int num;//本块号  
 int index;//下一块索引 0 表示空闲没有分配 -1表示结束 其他小于128，就是以及分配 表示下一块的块号  
 Object object;//表示当前块所占据的类  
 boolean begin;//用于表示此块是否为第一块  
 public Block(int num,int index,Object object,boolean begin)  
 {  
 将对应的数据修改  
 }  
 void setObject(Object object)  
 {  
 this.object=object;  
 }  
 public void BlockChange(int index,Object object,boolean begin)//第一个数据是本磁盘块所对应的下一个磁盘块的索引，第二个是类，第三个就是表示这个是不是第一块  
 {  
 修改index object begin三个数据  
 }  
  
}

5.文件分配表模块fat()

文件分配表类 FAT {  
 public int[] IndexArray;  
 public FAT()  
 {

初始化FAT表的数据  
 IndexArray=new int[128];  
 for(int i=0;i<128;i++)  
 {  
 if(i<=2)  
 {  
 IndexArray[i]=-1;  
 }  
 else  
 IndexArray[i]=0;  
 }  
 }  
}

6.已打开文件模块fileopened()

在Disk中的fileopened表ArrayList<FileTable> filesOpened;//记录文件是否打开的列表

Fileopened表中的FileTable类 FileTable {  
 String path;//文件路径名  
 int type;//文件属性  
 int num;//起始盘块号  
 int length;//文件长度  
 int operatetype;//操作类型  
 int begin;//读指针  
 int end;//写指针  
 public FileTable(String path,int type,int num,int length,int operatetype,int begin,int end)  
 {  
 初始化所有的数据  
 }  
  
}

1. 创建文件模块createFile()

public void createFile(){  
 Pane filePane = new Pane();  
 ImageView view = new ImageView();  
 Image icon = new Image("Txt.png");//用Txt图片作为文件类的图标  
 view.setImage(icon);  
 //添加图片，设置图片位置  
  
 HBox hb1 = new HBox();  
 Label l1 = new Label("请输入文件名");  
 hb1.getChildren().add(l1);  
 //设置请输入文件名标签的位置

HBox hb2 = new HBox();  
 TextField f1 = new TextField();  
 hb2.getChildren().add(f1);  
 //设置输入文本框的位置

Button bt1 = new Button("确定");  
 //设置确定按钮的位置

Button bt2 = new Button("取消");  
 //设置取消按钮的位置

filePane.getChildren().addAll(view,hb1,hb2,bt1,bt2);  
 Scene fileScene = new Scene(filePane,270,200);//设置窗口的大小  
 Stage fileStage = new Stage();  
 fileStage.setScene(fileScene);  
 fileStage.setTitle("新建文件");//设置窗口里面的人的  
 fileStage.show();

//设置当确定按钮被点击时所执行的函数  
 bt1.addEventFilter(MouseEvent.MOUSE\_CLICKED, new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent mouseEvent) {  
 if(mouseEvent.getButton()!=MouseButton.PRIMARY)  
 return;  
 String filename=f1.getText();//获取文件名  
 if(文件名中出现错误的写法)  
 {  
 ErrorWindows("新建文件错误","文件名中字符错误");  
 return;  
 }  
 在左侧的treeview中添加相应的treeitem  
 int index=FileSub.findFAT(3);//找到对应的空闲的盘块  
 FileSub.findFolder(printcurrent());//修改当前路径下的所在的文件夹  
 File X=new File(filename,printcurrent()+"\\"+filename,index,FileSub.F,0,1);// 新建一个File文件类  
 boolean can=FileSub.F.addChildrenNode(X);//当前的文件夹数据添加文件节 点，并更新数据  
 if(!can)  
 {  
 Label label=new Label("文件夹内容不能超过8个");  
 Writebox.getChildren().add(label);  
 AnchorPane.setTopAnchor(label,15.0+((Writebox.getChildren().size()-3))\*20);  
 fileStage.close();//点击确定后关闭界面  
 return;  
 }  
 //更改对应的磁盘块的内容  
 //修改FAT表  
   
 //在桌面设置文件的图片  
 Image FilePicture=new Image("Txt.png");  
 ImageView FP=new ImageView();  
 FP.setImage(FilePicture);  
 FP.setFitWidth(40);  
 FP.setFitHeight(40);  
 Label label=new Label(filename);  
 mainInterface.getChildren().addAll(FP,label);  
 设置图片图标的位置

//设置当图片被点击时所执行的函数  
 FP.addEventFilter(MouseEvent.MOUSE\_CLICKED, new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent mouseEvent) {  
 if(mouseEvent.getButton()==MouseButton.SECONDARY)  
 {  
 FileMenu X=new FileMenu();  
 X.show(FP,Side.BOTTOM,0,0);  
 }  
 }  
 });  
 fileStage.close();//点击确定后关闭界面  
 //对FAT表和磁盘的使用进行更新  
 }  
 });

设置取消按钮被点击后所发生事件  
 bt2.addEventFilter(MouseEvent.MOUSE\_CLICKED, new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent mouseEvent) {  
 if(mouseEvent.getButton()!=MouseButton.PRIMARY)  
 return;  
 fileStage.close();//关闭窗口  
 }  
 });  
}

1. 文件夹创建模块createfolder()

public void createFolder(){  
 Pane folderPane = new Pane();  
 Image icon1 = new Image("File.png");  
 ImageView view1 = new ImageView();  
 view1.setImage(icon1);  
 //设置图片的位置  
 HBox hb1 = new HBox();  
 Label l1 = new Label("请输入文件夹名");  
 hb1.getChildren().add(l1);  
 HBox hb2 = new HBox();  
 TextField f1 = new TextField();  
 hb2.getChildren().add(f1);  
 //设置文本输入框位置  
 Button bt1 = new Button("确定");  
 //设置确定按钮的位置  
  
 Button bt2 = new Button("取消");  
 //设置取消按钮的位置  
 folderPane.getChildren().addAll(view1,hb1,hb2,bt1,bt2);  
 Scene folderScene = new Scene(folderPane,270,200);  
 Stage folderStage = new Stage();  
 folderStage.setScene(folderScene);  
 folderStage.setTitle("新建文件夹");  
 folderStage.show();  
 bt1.addEventFilter(MouseEvent.MOUSE\_CLICKED, new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent mouseEvent) {  
 if(mouseEvent.getButton()!=MouseButton.PRIMARY)  
 return;  
 String filename=f1.getText();  
 if(文件夹名称中有错误字符)  
 {  
 ErrorWindows("新建文件夹错误","文件夹名中字符错误");  
 }  
 新增一个Treeitem节点加入ROOT中  
 FileSub.mkpathdir(filename,printcurrent());  
 //对后端数据的修改

Image FilePicture=new Image("File.png");  
 ImageView FP=new ImageView();  
 FP.setImage(FilePicture);  
 FP.setFitWidth(40);  
 FP.setFitHeight(40);  
 Label label=new Label(filename);  
 mainInterface.getChildren().addAll(FP,label);  
设定桌面图标和标签所在位置

FP.addEventFilter(MouseEvent.MOUSE\_CLICKED, new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent mouseEvent) {  
 if(mouseEvent.getButton()==MouseButton.SECONDARY)  
 {  
 右键点击时出现选择框  
 }  
 }  
 });  
 folderStage.close();//关闭窗口  
 changeDiskusing();  
 changeFAT();//关闭窗口后更新  
 }  
 });  
  
 bt2.addEventFilter(MouseEvent.MOUSE\_CLICKED, new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent mouseEvent) {  
 点击取消时关闭界面  
 }  
 });  
}

1. 命令行删除文件(夹)操作模块delete\_pathfile()

public static boolean delete\_pathfile(String path)  
{  
 File file=findfile(path);  
 int pos=0;  
 for(int index=file.num;index!=-1;index=pos)  
 {  
 不断的迭代然后更新磁盘块 ，归还磁盘块，更新FAT表  
 }  
 for(int i=0;i<file.parent.item.size();i++)  
 {  
 if(文件的某个父节点中有对应要删除的节点) {  
 对父节点的数据进行相应的修改

return true;  
 }  
 }  
 return false;  
}

1. 文件读写或只读打开模块 openfile()

public void openfile(File X) {  
 if(需要打开的文件不存在) {  
 在命令行一栏中告知使用者  
 }  
 else {  
 打开文件，并显示文件中的内容  
 if(文件是可读写文件)  
 {  
 设置窗口的标题，加上读写二字  
 }else{  
 设置窗口的标题，加上只读二字  
 }  
 fileStage.setOnCloseRequest(new EventHandler<WindowEvent>() {  
 @Override  
 public void handle(WindowEvent windowEvent) {  
  
 try {  
 当关闭文件后，更新文件的内容，

磁盘块数据，FAT表  
 } catch (UnsupportedEncodingException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 });  
 fileStage.show();  
 }  
  
}

1. 文件夹打开模块 openfolder()

public void openfolder(Folder folder)  
{  
 Pane filePane=new Pane();  
 for(int i=0;i<folder.children.size();i++)  
 {  
 if(子节点属于文件类)  
 {  
 Image fl=new Image("Txt.png");  
 ImageView FIV=new ImageView();

Label FLname=new Label(((File)folder.children.get(i)).fileName);  
 在新打开的界面中添加图标和标签

//如果图标被点击  
 FIV.addEventFilter(MouseEvent.MOUSE\_CLICKED, new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent mouseEvent) {  
 if(mouseEvent.getClickCount()==2)  
 执行openfile函数  
 }  
 });  
 }  
 else if(子节点属于文件夹类){  
 Image fl=new Image("File.png");  
 ImageView FIV=new ImageView();  
 Label FLname=new Label(((Folder)folder.children.get(i)).folderName);  
 在新打开的界面中添加图标和标签

//如果图标被点击  
 FIV.addEventFilter(MouseEvent.MOUSE\_CLICKED, new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent mouseEvent) {  
 执行openfolder函数  
 }  
 });  
 }  
 //设置文件图片和标签的位置  
 }  
 Scene fileScene = new Scene(filePane,600,400);//设置窗口的大小  
 Stage fileStage = new Stage();  
 fileStage.setScene(fileScene);  
 fileStage.setTitle(folder.folderName);//设置窗口的标题  
 fileStage.show();  
  
}