

模拟文件管理系统-设计方案报告

学号	姓名	课号
1852137	张艺腾	42036901

目录

模拟文件管理系统-设计方案报告

目录	
项目需求	
基本任务	
功能描述	
项目目的	
开发环境	
项目结构	
系统分析	
显示链接法	
位图、FAT表	
系统设计及实现	
组件设计	
功能实现	
创建文件	
删除文件	
重命名	
打开文件	
格式化	
功能展示及详细操作说明	
创建文件	
创建文件夹	
格式化	
删除文件夹	
删除文件	
重命名	
打开文件	
搜索文件	
作者	

项目需求

基本任务

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。

功能描述

- 文件存储空间管理可采取显式链接（如FAT）或者其他方法。（即自选一种方法）
- 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。
- 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

项目目的

- 熟悉文件存储空间的管理；
- 熟悉文件的物理结构、目录结构和文件操作；
- 熟悉文件系统管理实现；
- 加深对文件系统内部功能和实现过程的理解

开发环境

- **开发环境:** Windows 10
- **开发软件:** Eclipse
- **开发语言:** JavaSE (jdk1.8.0_241)
- **开发工具包:** Swing

项目结构

```
1 | MyFileSystem.exe
2 | MyFileSystem.jar
3 | README.md
4 | README.pdf
5 | 模拟文件管理系统-设计方案报告.pdf
6 |
7 |─myFileSystem（模拟系统文件夹，请勿删除）
8 | |   ReadMe.txt
9 | |
10 | |─1
11 | |   1BitMap&&Fat.txt
12 | |   recover.txt
13 | |
14 | |─2
15 | |   2BitMap&&Fat.txt
16 | |   recover.txt
17 | |
18 | |─3
19 | |   3BitMap&&Fat.txt
20 | |   recover.txt
21 | |
22 | |─4
```

```
23 | | 4BitMap&&Fat.txt
24 | | recover.txt
25 | |
26 | |─5
27 | | 5BitMap&&Fat.txt
28 | | recover.txt
29 | |
30 | |─6
31 | | 6BitMap&&Fat.txt
32 | | mojito.txt
33 | | recover.txt
34 | |
35 | |─7
36 | | 7BitMap&&Fat.txt
37 | | recover.txt
38 | |
39 | |─8
40 | | 8BitMap&&Fat.txt
41 | | recover.txt
42 | |
43 | |└9
44 | | 9BitMap&&Fat.txt
45 | | recover.txt
46 | |
47 |└src
48 |   |─image
49 |   | document.png
50 |   | folder.png
51 |   |
52 |   |─Java
53 |   | Folder.java
54 |   | myFiles.java
55 |   | MyFileSystem.java
56 |   | tableModel.java
57 |   |
58 |   |└myFileSystem
```

系统分析

显示链接法

本文件系统中, 文件存储空间管理使用**显示链接**的方法, 文件中的内容存放在磁盘不同的块中, 每次创建文件时为文件分配数量合适的空闲块。每次写文件时按顺序将文件内容写在相应块中; 删除文件时将原先有内容的位置置为空即可。

位图、FAT表

磁盘空闲空间管理在**位图**的基础上进行改造, 将存放磁盘上文件位置信息的**FAT表**与传统的位图进行结合, 磁盘空闲的位置使用 0 标识, 放有文件的盘块存放文件所在的下一个盘块的位置, 文件存放结束的盘块位置使用 1 标识。

系统设计及实现

组件设计

- 文件树: `Java.Swing.JTree`
- 文件表: `Java.Swing.JTable`
- 搜索板块
 - 搜索框: `Java.Swing.JTextField`
 - 提示语: `Java.Swing.JLabel`
 - 按钮: `Java.Swing.JButton`
- 详细信息展示条: `Java.Swing.JLabel`

功能实现

创建文件

通过调用 `File` 类的 `createFile()` 来实现, 创建后刷新FAT和位图, 如果虚拟存储空间不够则会提示创建失败

```

1 JMenuItem createFileItem = new JMenuItem("新建文件");
2 createFileItem.addActionListener(new ActionListener() {
3     @Override
4     public void actionPerformed(ActionEvent e) {
5         DefaultMutableTreeNode node =(DefaultMutableTreeNode)
6             tree.getLastSelectedPathComponent();
7         myFiles temp = (myFiles) node.getUserObject();
8         int blokName = temp.getBlockName();
9         Folder currentBlock = folders.get(blokName - 1);
10
11         String inputValue;
12         double capacity;
13
14         JOptionPane inputPane = new JOptionPane();
15         inputPane.setPreferredSize(new Dimension(600, 600));
16         inputPane.setInputValue(JOptionPane.showInputDialog("文件名: "));
17         if (inputPane.getInputValue() == null) {
18             return;
19         }
20         inputValue = inputPane.getInputValue().toString();
21         inputPane.setInputValue(JOptionPane.showInputDialog("文件大小
22         (KB):"));
23         if (inputPane.getInputValue() == null) {
24             return;
25         }
26         capacity =
27         Double.parseDouble(inputPane.getInputValue().toString());
28
29         File newFile = new File(temp.getFilePath() + File.separator +
30         inputValue + ".txt");
31         if (!newFile.exists() && !inputValue.equals(null)) {
32             try {
33                 if (currentBlock.createFile(newFile, capacity)) {
34                     DefaultMutableTreeNode newNode = new
35                     DefaultMutableTreeNode(
36                         new myFiles(newFile, blokName, capacity));
37                     model.removeRows(0, model.getRowCount());
38                     model.addRow(new myFiles(newFile, blokName, capacity));
39                     fileTable.updateUI();
40                     upDateBlock(currentBlock);

```

```

37         JOptionPane.showMessageDialog(null, "创建成功! 请刷新文件夹!", "Success", JOptionPane.DEFAULT_OPTION);
38     }
39     } catch (IOException e1) {
40         JOptionPane.showMessageDialog(null, "创建失败!!!", "Error",
JOptionPane.ERROR_MESSAGE);
41     }
42 }
43 }
44 });
45 myMenu.add(createFileItem);

```

删除文件

通过调用 File 类的 delete() 来实现, 对于文件夹, 需要先递归调用删除函数将文件删除后再删文件夹本身, 删除成功后刷新FAT和位图

```

1 JMenuItem deleteItem = new JMenuItem("删除");
2 deleteItem.addActionListener(new ActionListener() {
3     @Override
4     public void actionPerformed(ActionEvent e) {
5         DefaultMutableTreeNode node = (DefaultMutableTreeNode)
6             tree.getLastSelectedPathComponent();
7         myFiles temp = (myFiles) node.getUserObject();
8         int blokName = temp.getBlockName();
9         Folder currentBlock = folders.get(blokName - 1);
10        int choose = JOptionPane.showConfirmDialog(null, "确定删除?",
"confirm", JOptionPane.YES_NO_OPTION);
11        if (choose == 0) {
12            if (currentBlock.deleteFile(temp.getMyFile(), temp.getSpace()))
13            {
14                try {
15                    currentBlock.rewriteBitMap();
16                    currentBlock.rewriteRecoverWriter();
17                } catch (IOException e1) {
18                    e1.printStackTrace();
19                }
20                upDateBlock(currentBlock);
21                JOptionPane.showMessageDialog(null, "删除成功, 请刷新文件夹!",
"Success", JOptionPane.DEFAULT_OPTION);
22            } else {
23                JOptionPane.showMessageDialog(null, "删除失败!!!", "Error",
JOptionPane.ERROR_MESSAGE);
24            }
25        }
26    });
27    myMenu.add(deleteItem);

```

重命名

通过调用File类的 renameTo() 方法进行重命名, 如果相同目录下有相同的文件, 则重命名将会失败。

```

1 JMenuItem renameItem = new JMenuItem("重命名");
2 renameItem.addActionListener(new ActionListener() {
3     @Override

```

```

4      public void actionPerformed(ActionEvent e) {
5          DefaultMutableTreeNode node = (DefaultMutableTreeNode)
6              tree.getLastSelectedPathComponent();
7          myFiles temp = (myFiles) node.getUserObject();
8          int blokName = temp.getBlockName();
9          Folder currentBlock = folders.get(blokName - 1);
10
11         String inputValue = null;
12         JOptionPane inputPane = new JOptionPane();
13         inputPane.setInputValue(JOptionPane.showInputDialog("新的文件名:"));
14         if (inputPane.getInputValue() == null) {
15             return;
16         }
17         inputValue = inputPane.getInputValue().toString();
18         try {
19             currentBlock.renameFile(temp.getMyFile(), inputValue,
20                 temp.getSpace());
21             JOptionPane.showMessageDialog(null, "重命名成功, 请刷新文件夹",
22                 "Success",
23                                     JOptionPane.DEFAULT_OPTION);
24         } catch (IOException e1) {
25             JOptionPane.showMessageDialog(null, "重命名失败!!!", "Error",
26                                     JOptionPane.ERROR_MESSAGE);
27         }
28     };
29     myMenu.add(renameItem);

```

打开文件

双击程序右侧显示面板中的文件即可打开对应文件，通过对鼠标操作加监听来实现

```

1  fileTable.addMouseListener(new MouseAdapter() {
2      @Override
3      public void mouseClicked(MouseEvent e) {
4          super.mouseClicked(e);
5          if (e.getClickCount() == 2 && e.getButton() == MouseEvent.BUTTON1)
6          {
7              String fileName = ((String)
8                  model.getValueAt(fileTable.getSelectedRow(), 0));
9              String filePath = ((String)
10                  model.getValueAt(fileTable.getSelectedRow(), 1));
11              try {
12                  if (Desktop.isDesktopSupported()) {
13                      Desktop desktop = Desktop.getDesktop();
14                      desktop.open(new File(filePath));
15                  }
16              } catch (IOException e1) {
17                  JOptionPane.showMessageDialog(null, "抱歉, 出了一些错误!",
18                      "Fail to open", JOptionPane.ERROR_MESSAGE);
19              }
20              JOptionPane.showMessageDialog(null, "File Name: " + fileName +
21                  "\n File Path: " + filePath,
22                      "content", JOptionPane.INFORMATION_MESSAGE);
23          }
24      }
25  };

```

格式化

先递归删除该目录下的所有文件，然后更新位图和FAT

```
1 JMenuItem formatItem = new JMenuItem("格式化");
2 formatItem.addActionListener(new ActionListener() {
3     @Override
4     public void actionPerformed(ActionEvent e) {
5         DefaultMutableTreeNode node = (DefaultMutableTreeNode)
6         tree.getLastSelectedPathComponent();
7         myFiles temp = (myFiles) node.getUserObject();
8         int blokName = temp.getBlockName();
9         Folder currentBlock = folders.get(blokName - 1);
10        int choose = JOptionPane.showConfirmDialog(null, "确定格式化文件夹
11        吗?", "confirm", JOptionPane.YES_NO_OPTION);
12        if (choose == 0) {
13            try {
14                if (temp.getMyFile().isDirectory()) {
15                    for (File myfile : temp.getMyFile().listFiles()) {
16                        currentBlock.deleteFile(myfile, getSpace(myfile));
17                    }
18                    updateBlock(currentBlock);
19                    JOptionPane.showMessageDialog(null, "格式化成功，请刷新文件
20                    夹!", "Success", JOptionPane.DEFAULT_OPTION);
21                    currentBlock.rewriteBitMap();
22                }
23            } catch (Exception E1) {
24                JOptionPane.showMessageDialog(null, "格式化失败!!!", "Error",
25                JOptionPane.ERROR_MESSAGE);
26            }
27        }
28    }
29 });
30 myMenu.add(formatItem);
```

功能展示及详细操作说明

创建文件

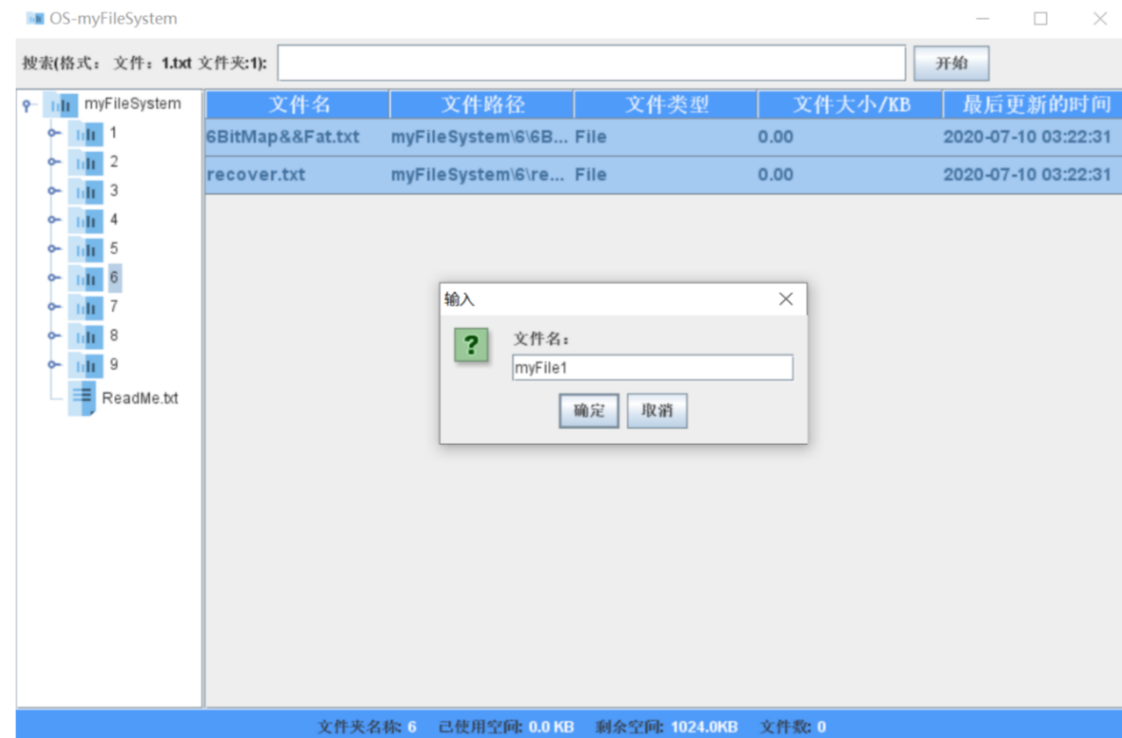
- 在**文件夹内**双击运行 `MyFilesystem.exe`，进入文件管理模拟系统如下图



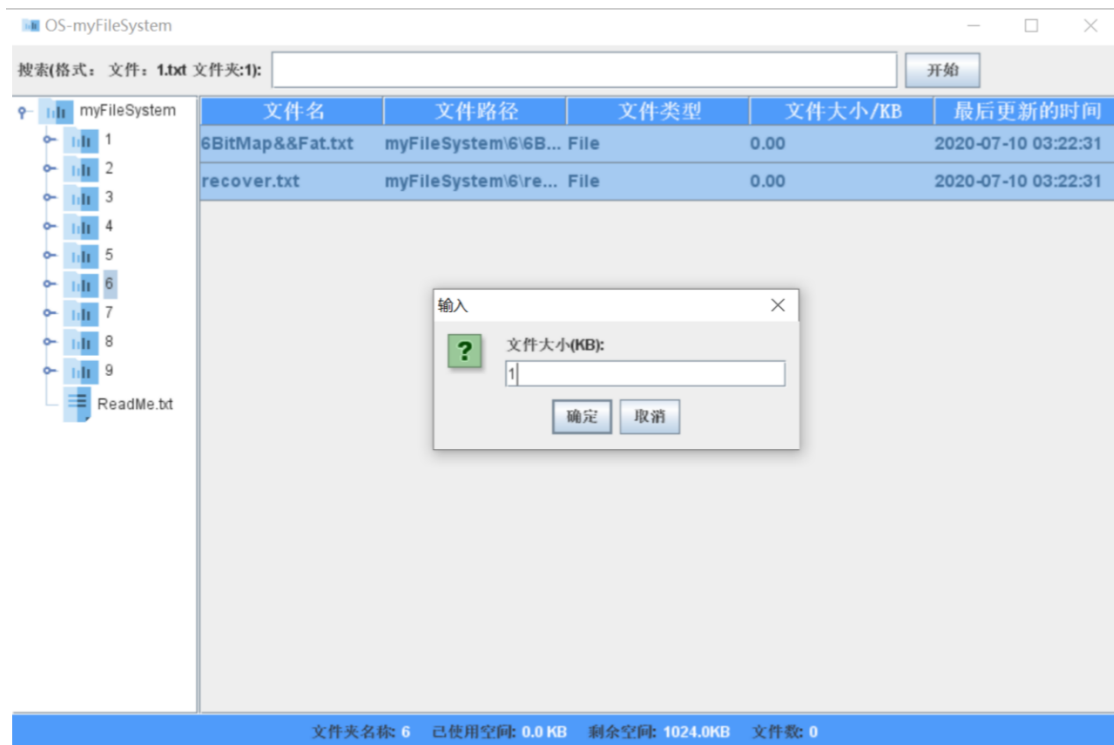
- 选中文件夹，右键显示操作列表，如下图所示



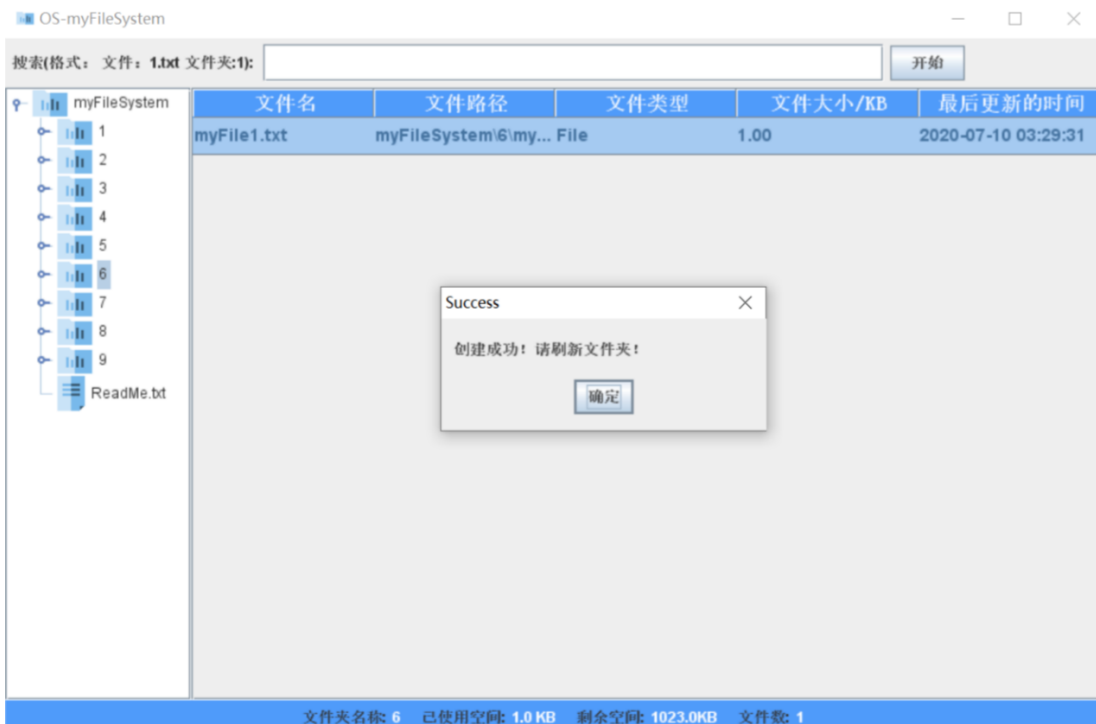
- 点击 新建文件夹，在弹出的提示框内输入文件名，然后点击 确定



- 输入文件大小，再次点击 **确定**

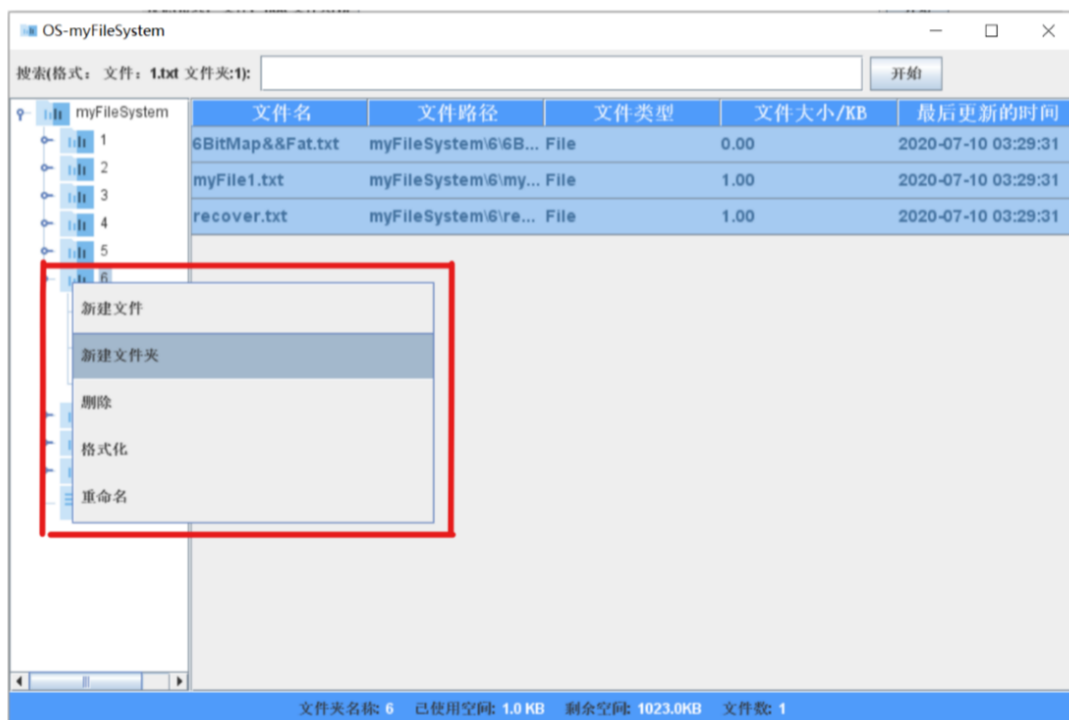


- 系统将提示新建文件成功，再次点击 **确定**

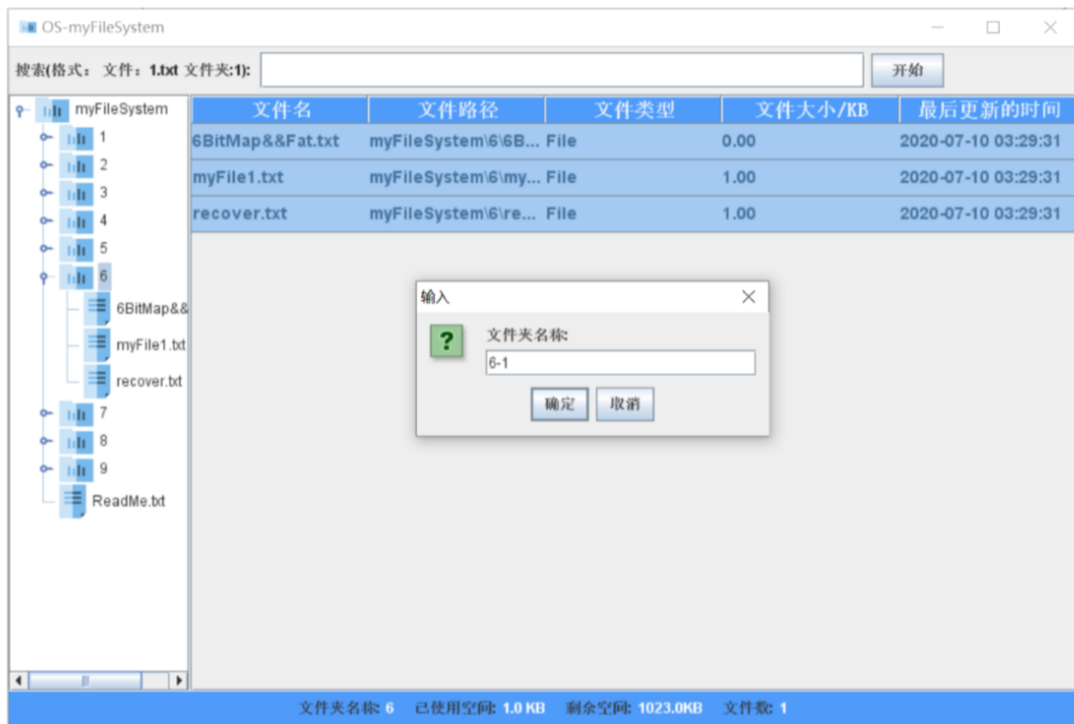


创建文件夹

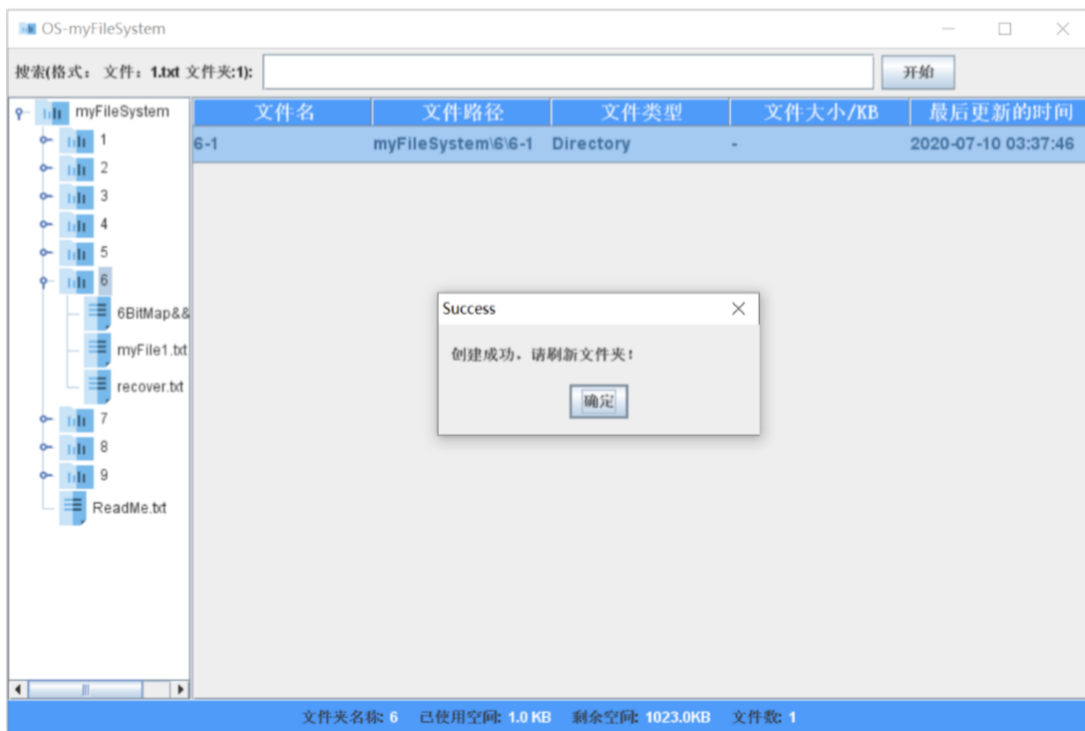
- 右键召出菜单列表, 点击 新建文件夹



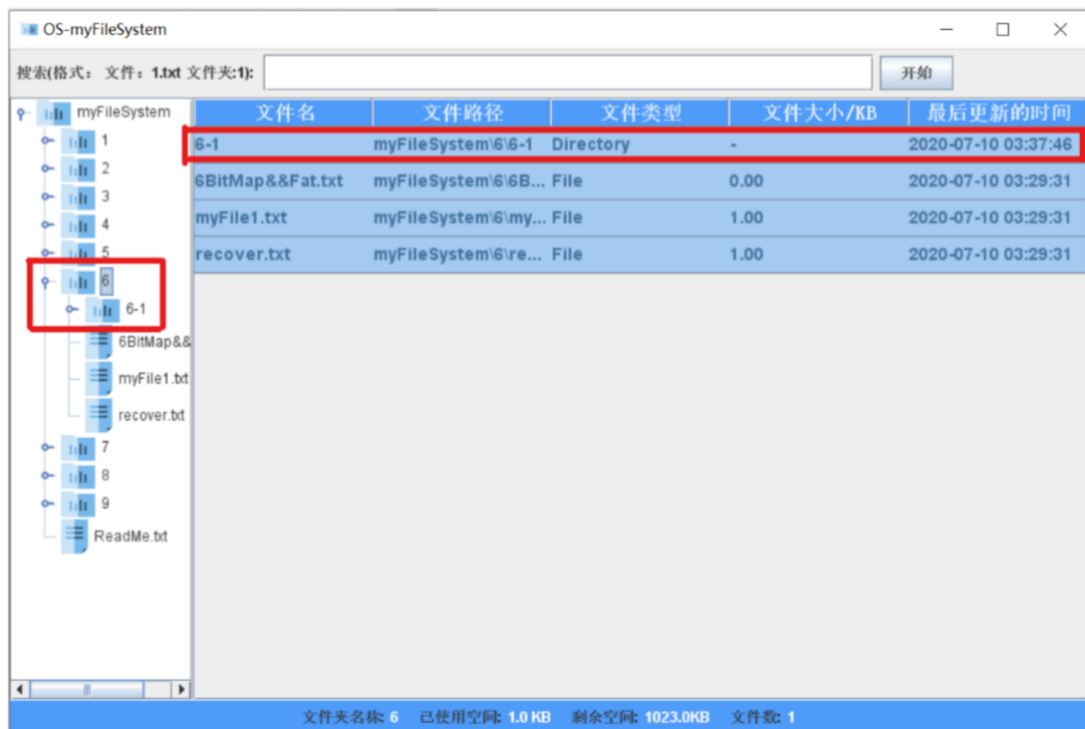
- 在弹出的提示框内输入文件夹名称, 然后点击 确定



- 系统提示创建成功，再次点击

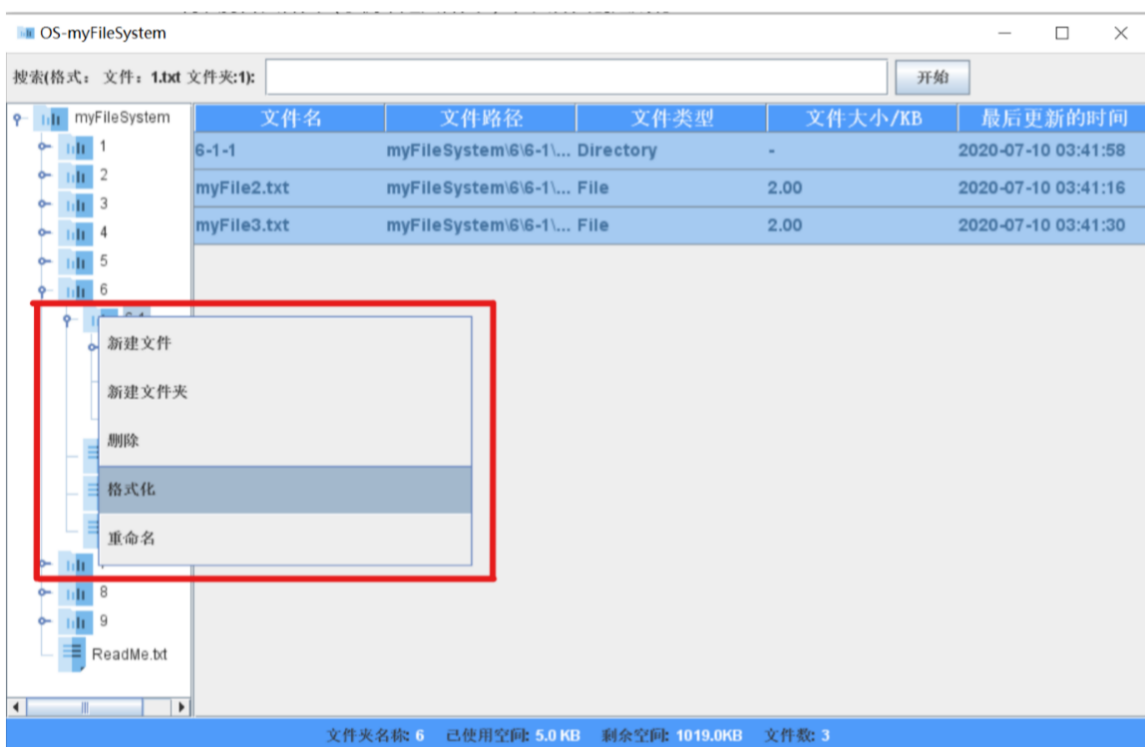


- 再次打开文件夹（示例中是文件夹6），文件夹创建成功

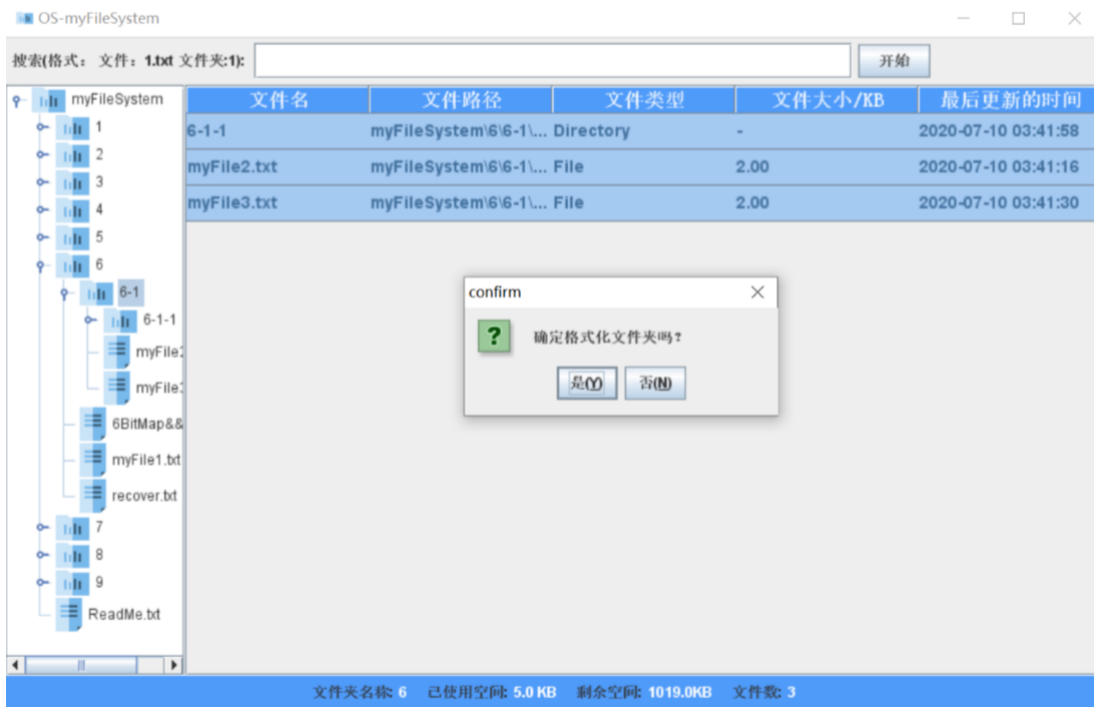


格式化

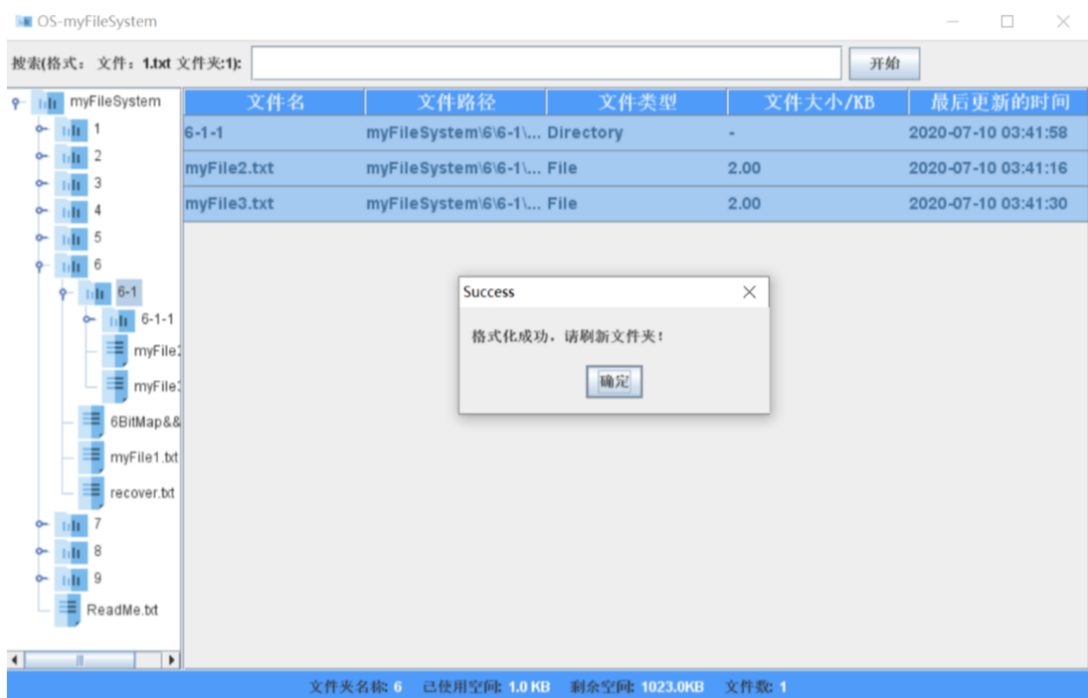
- 选定要格式化的文件夹，右键召出菜单列表，点击 格式化



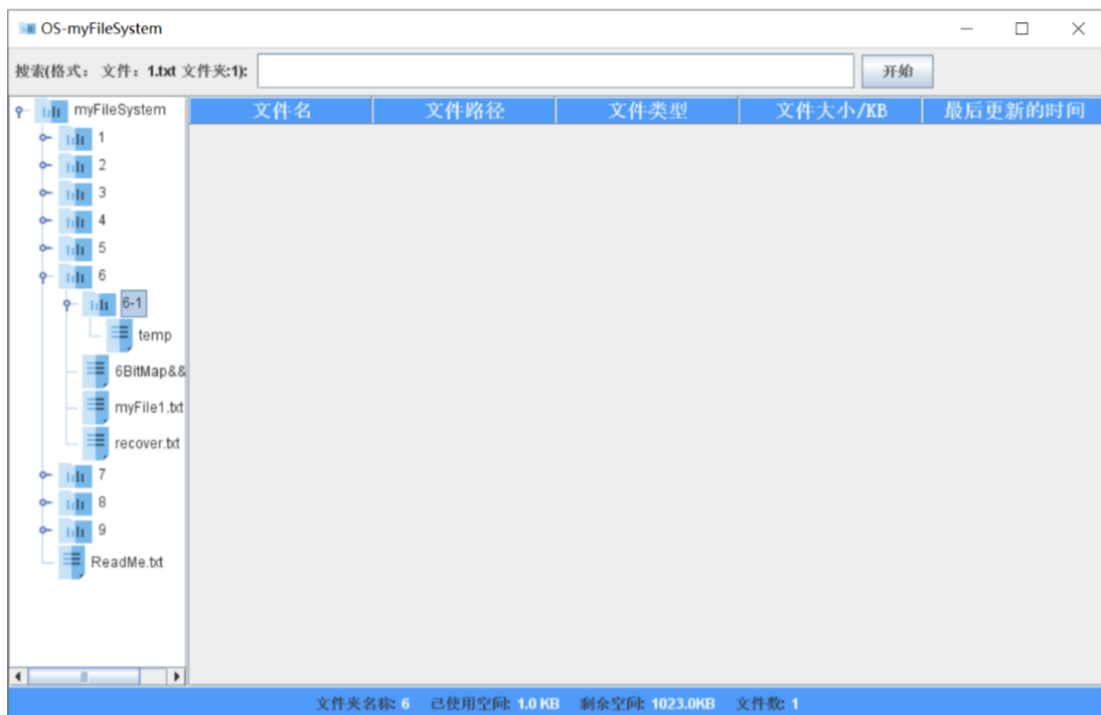
- 在弹出的对话框中选择 是 或 否



- 系统提示格式化成功，再次点击 确定



- 再次打开格式化后的文件夹，格式化成功

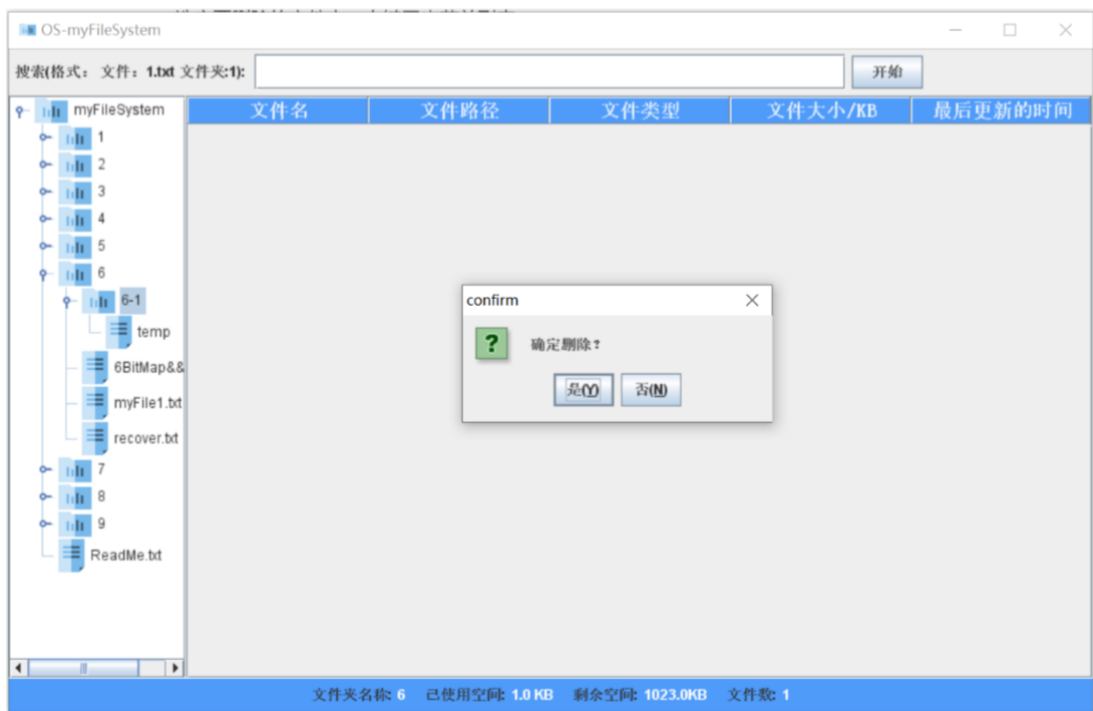


删除文件夹

- 选定要删除的文件夹，右键召出菜单列表



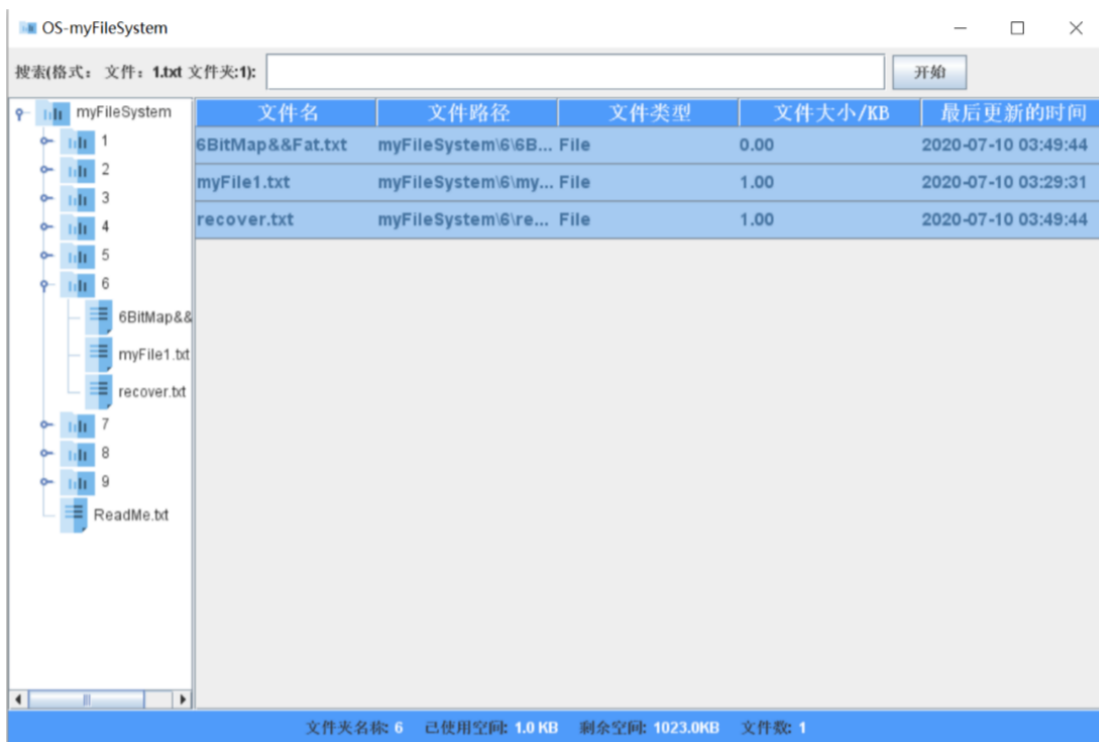
- 在弹出的对话框中选择是或否



- 系统弹出对话框提示删除成功



- 再次打开文件夹, 删除成功

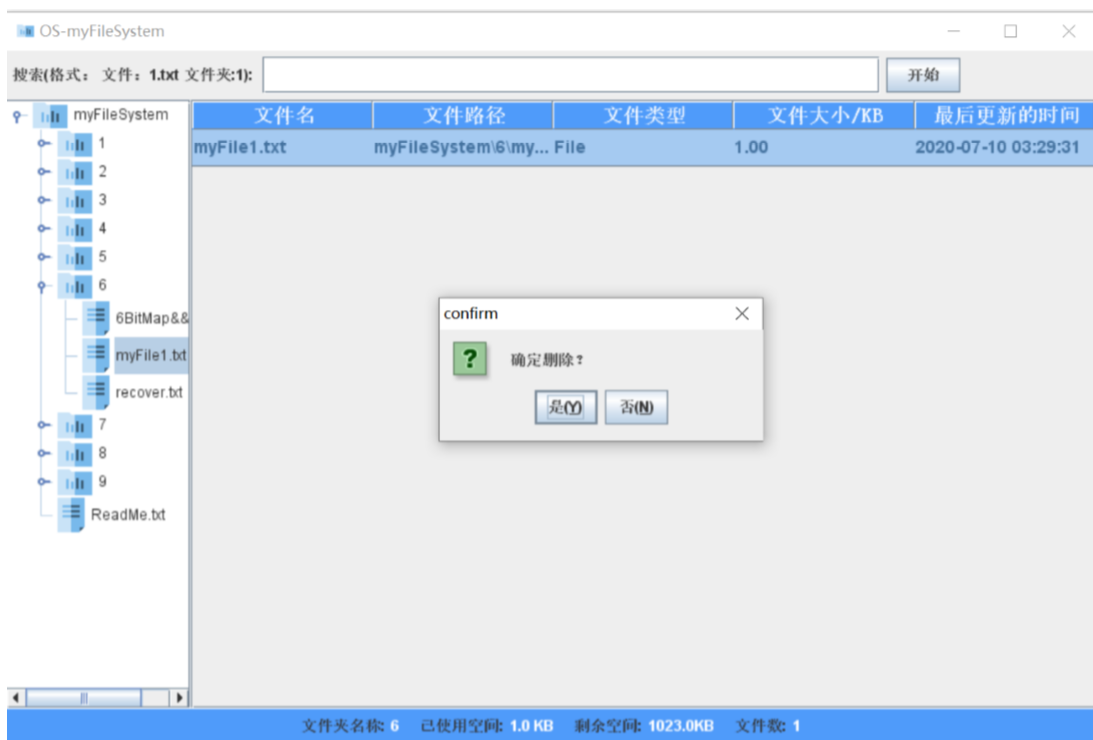


删除文件

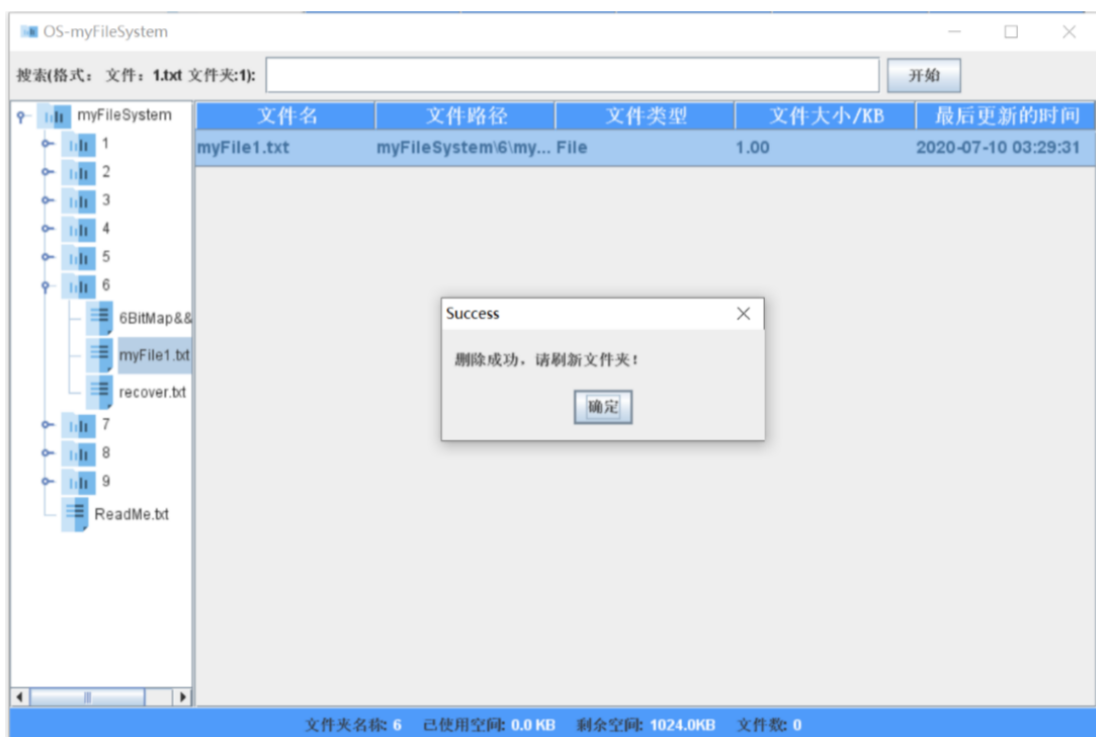
- 选定要删除的文件，右键召出菜单列表，点击 **删除**



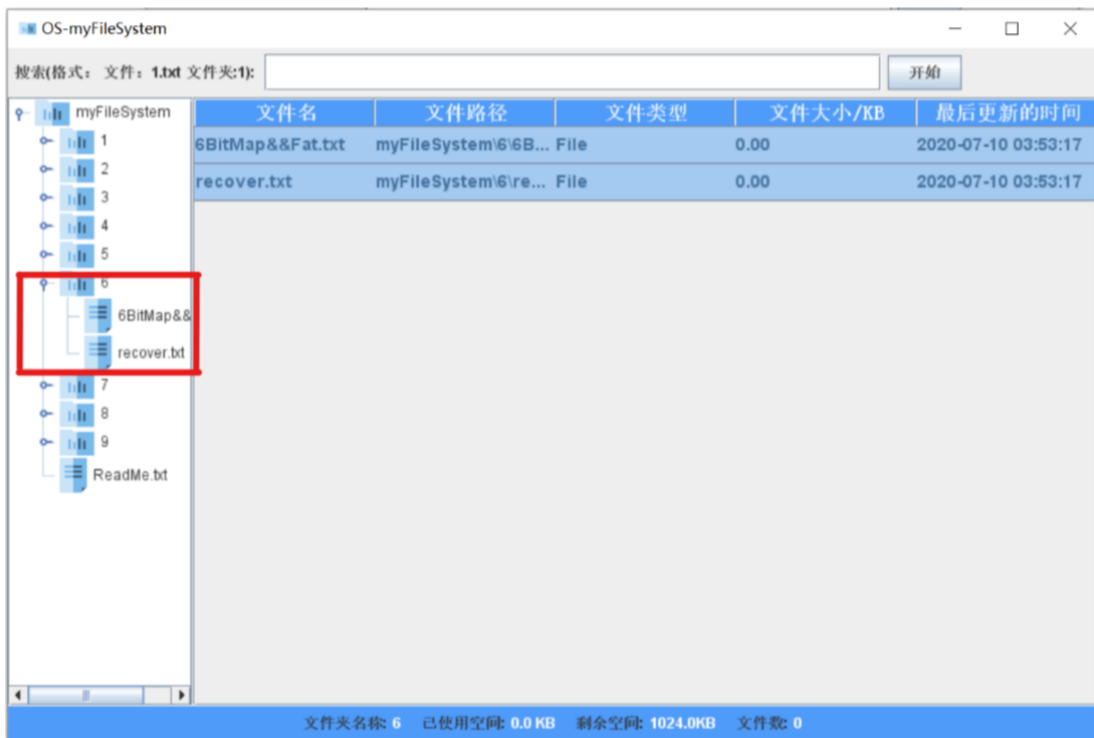
- 在弹出的系统对话框中选择 **是** 或 **否**



- 选 是 则系统提示删除成功



- 重新打开文件夹, 删除成功

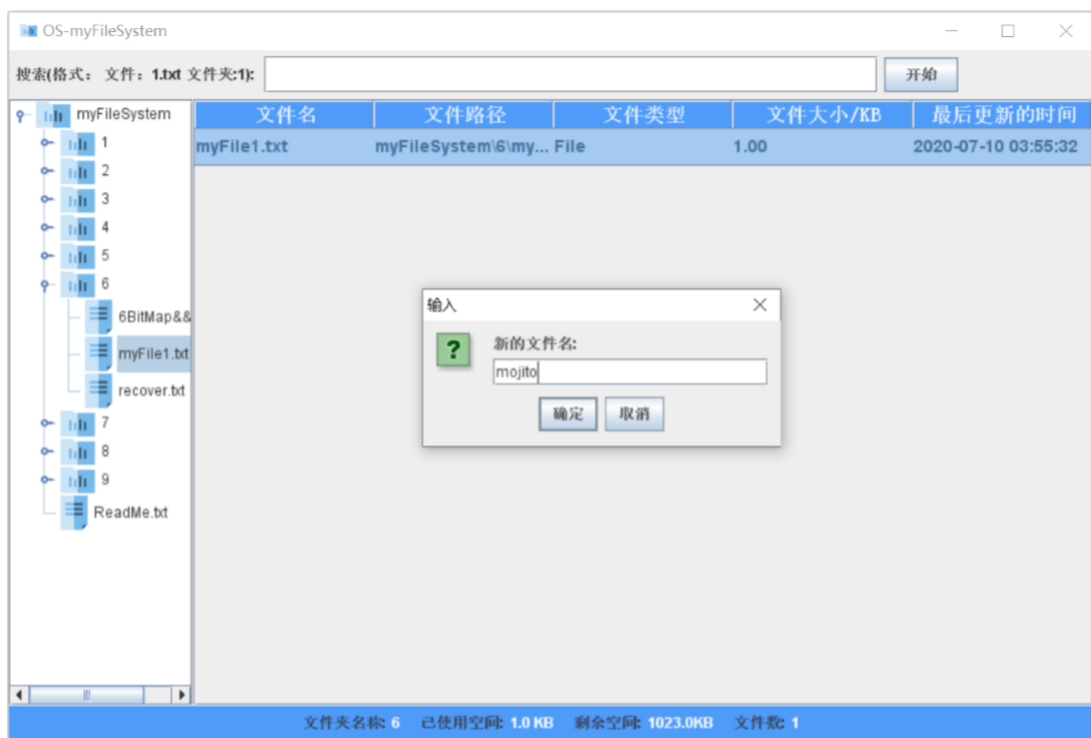


重命名

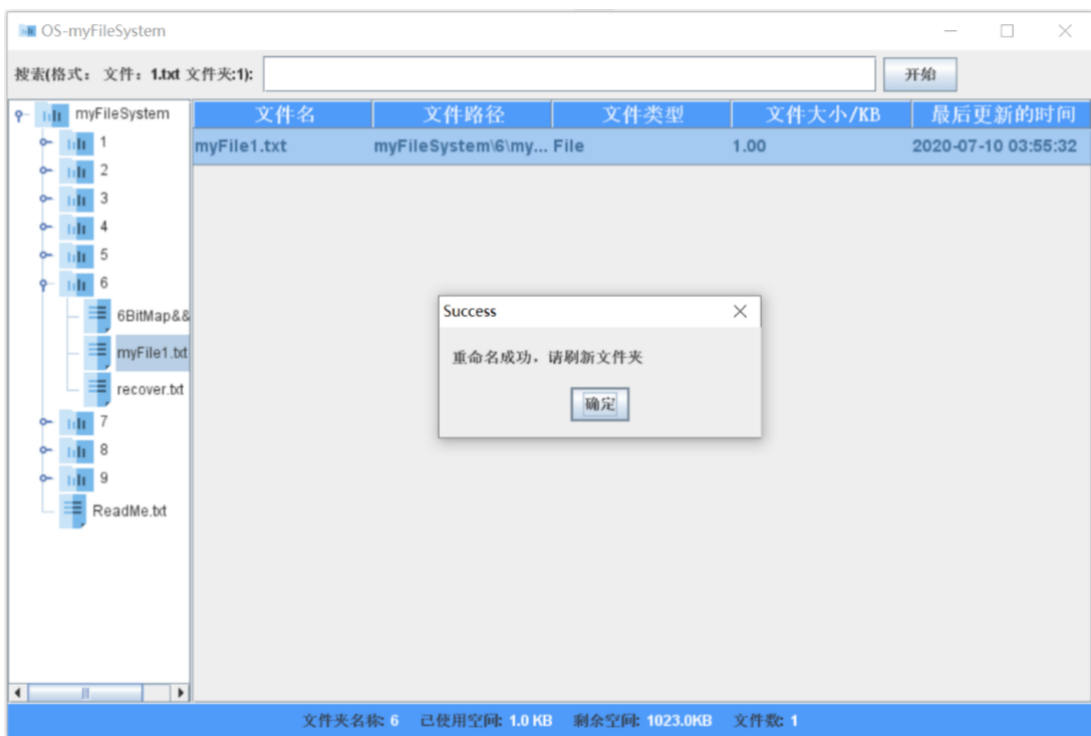
- 选中要重命名的文件或文件夹，右键召出菜单列表，选择重命名



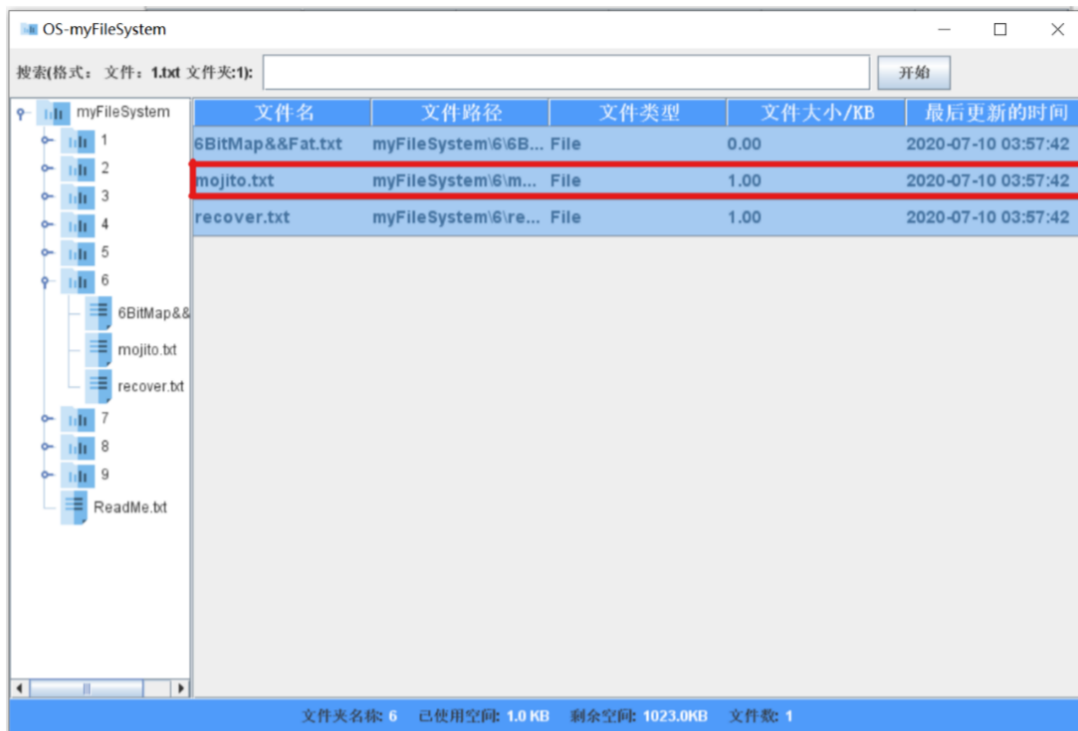
- 在弹出的对话框中填上要修改的文件名，然后点击 确定



- 系统弹出对话框提示重命名成功

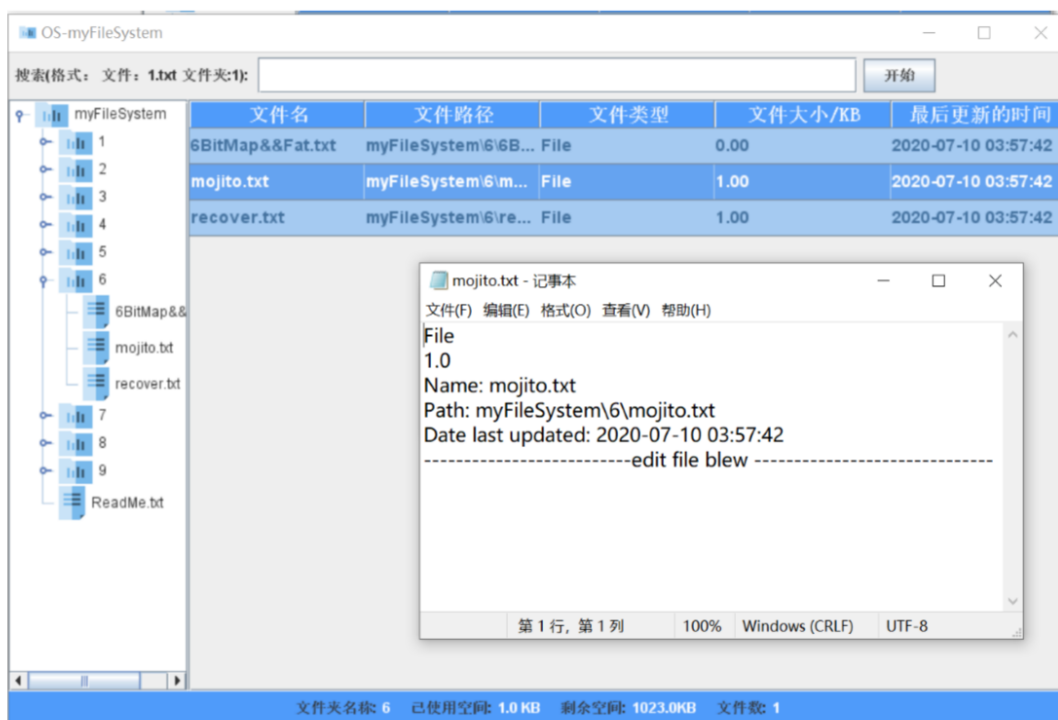


- 再次打开文件夹, 重命名成功



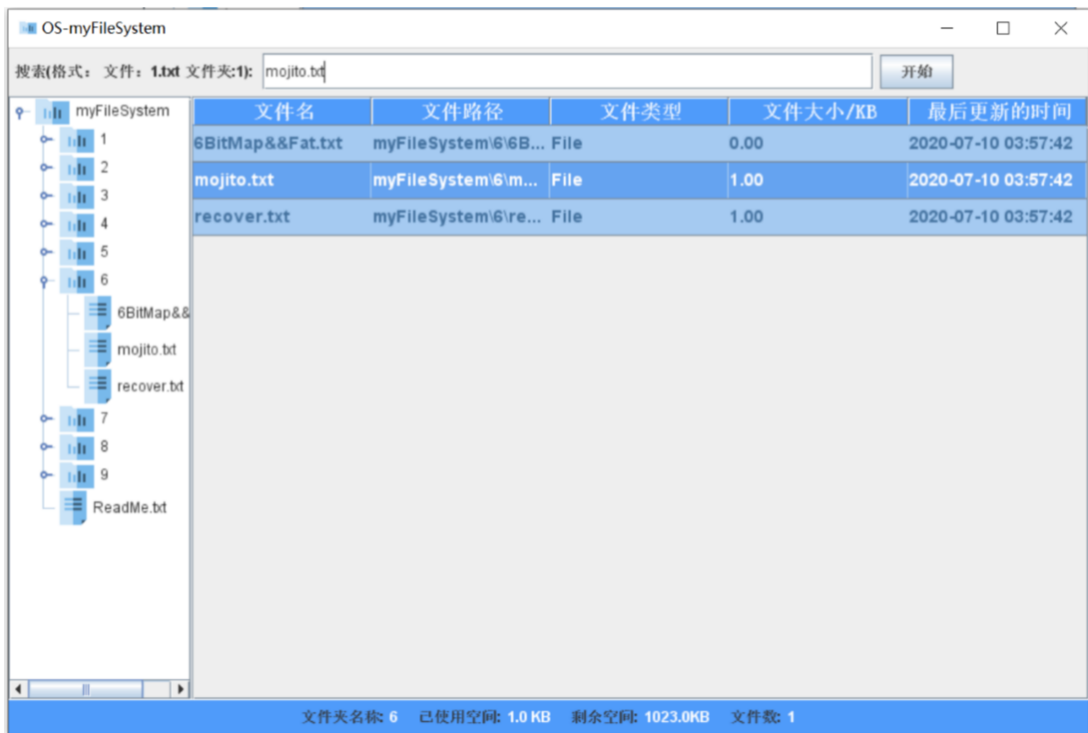
打开文件

- 找到要打开的文件，双击文件表中的文件

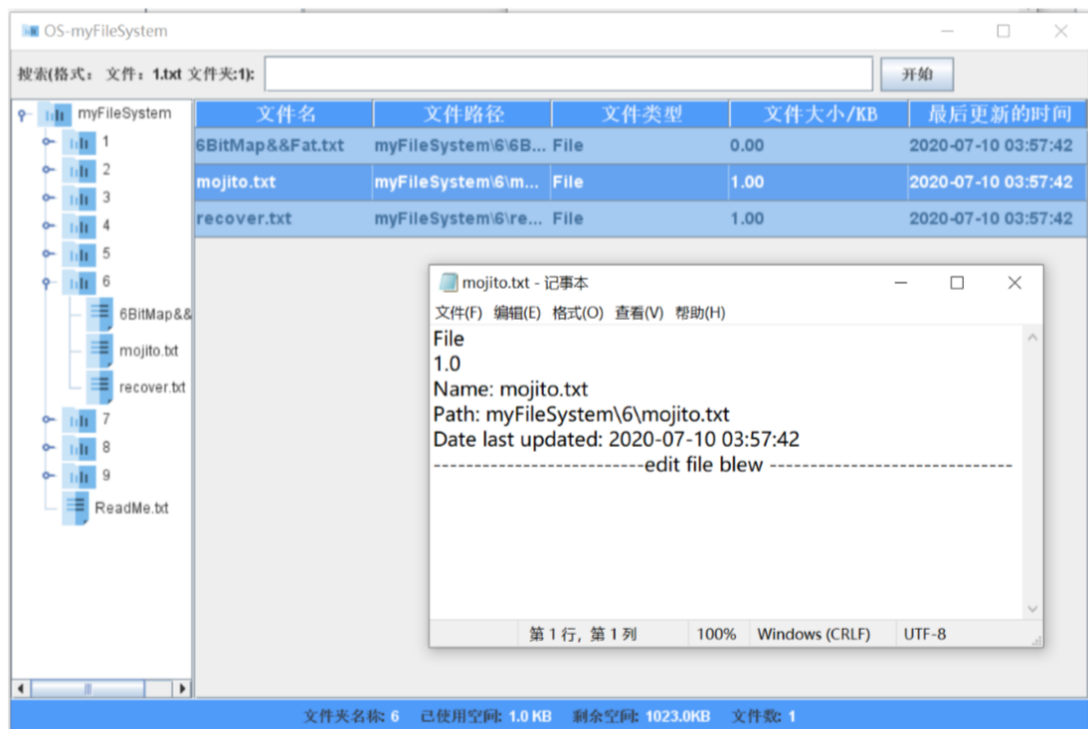


搜索文件

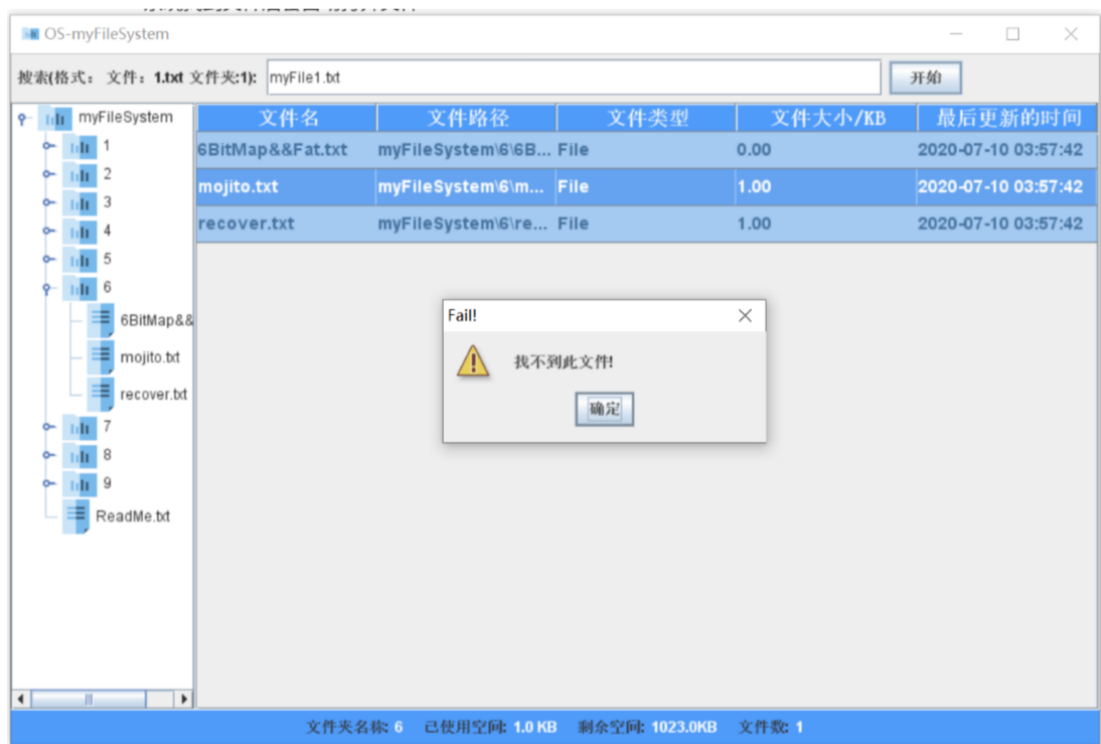
- 在上方的搜索框中输入要搜索的文件或文件夹名称（注：文件名称格式为 myFile.txt 文件夹名称格式为 6-1）然后点击 开始



- 系统找到文件后会自动打开文件



- 系统找不到文件则会弹出对话框提示查找失败



作者

姓名: Kerr

联系方式: email:kerr99801@gmail.com