

# 文件管理系统

1452764 何冬怡

## 目录

|                                  |          |
|----------------------------------|----------|
| 项目需求 .....                       | 3        |
| <b>使用手册 .....</b>                | <b>3</b> |
| 1. 进入系统 .....                    | 3        |
| 2. 浏览文件——ls .....                | 3        |
| 3. 跳转当前位置——cd .....              | 4        |
| 4. 新建文件夹/文件——mkdir, mkfile ..... | 4        |
| 5. 编辑和打开文档——edit, open .....     | 4        |
| 6. 删除和格式化——rm, rmall .....       | 4        |
| 7. 获取帮助 .....                    | 5        |
| 8. 退出系统 .....                    | 5        |
| 程序特性 .....                       | 6        |
| 9. 程序环境 .....                    | 6        |
| 10. 底层结构 .....                   | 6        |
| 11. 前端交互 .....                   | 6        |
| 12. 权限管理 .....                   | 6        |
| 13. 报错管理 .....                   | 6        |
| 14. 文件目录 .....                   | 6        |
| 成员变量/函数接口 .....                  | 7        |

## 项目需求

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。

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。

## 使用手册

### 1. 进入系统

打开 exe 文件后，出现欢迎界面和指示信息。输入任意用户名(无需密码)进入系统。

```
=====
                                欢迎使用文件管理系统
                                1452764 何冬怡
=====
本系统使用类Linux命令，支持下列命令
ls <path>      列出当前目录下所有文件/文件夹
cd <path>      跳转到目标路径，支持. 回到上一级, 绝对路径及相对路径
mkdir <name>    在当前目录下新建文件夹
mkfile <name>   在当前目录下新建文档
rm <name>       删除当前目录下的文件/文件夹
edit <name>     编辑文件内容
open <name>     打开文件内容
rmall          系统格式化
help           获取帮助
quit           退出系统
=====
Please input your user name: _

Please input your user name: donggu
[donggu @ /]#
```

命令行前方句柄，@前面为当前用户名，@后面为当前目录（"/"为根目录）。

### 2. 浏览文件——ls

使用 ls 指令查看文件目录。和 linux 系统相仿，直接输入 ls，显示当前目录下的文件/文件夹信息，包括名字，类型，创建者，大小，创建时间和最后修改时间等信息；

```
[donggu @ /]# ls
name          type    owner   size   createTime      lastUpdate
publicFile    <dir>   public  0      2016/6/22 2:32:25 2016/6/22 2:32:25
```

ls 后面也可以带上文件夹路径以展示目标路径下文件信息。参数路径支持绝对路径和相对路径，包括..返回上级，.指代当前目录等。

```
[donggu @ /]# ls publicFile
name          type    owner   size   createTime      lastUpdate
hello         <dir>   donggu  0      2016/6/22 20:56:42 2016/6/22 20:56:42
world         <dir>   donggu  0      2016/6/22 20:56:50 2016/6/22 20:56:50
```

### 3. 跳转当前位置——cd

利用 cd 指令跳转当前位置，参数同样支持绝对路径和相对路径。可在命令行首看见当前目录的变化。

```
[donggu @ /]# cd /publicFile/hello
[donggu @ hello]#
```

### 4. 新建文件夹/文件——mkdir, mkfile

使用 mkdir 新建文件夹，用 mkfile 新建文件，需要在指令后面加上文件/文件夹名。也支持绝对路径和相对路径。

```
[donggu @ hello]# mkdir /dir1
[donggu @ hello]# mkfile hhh
[donggu @ hello]# ls /
name          type    owner   size   createTime      lastUpdate
publicFile    <dir>   public  0      2016/6/22 2:32:25 2016/6/22 21:01:16
dir1          <dir>   donggu  0      2016/6/22 21:01:09 2016/6/22 21:01:09
[donggu @ hello]# ls
name          type    owner   size   createTime      lastUpdate
hhh           <file>  donggu  0      2016/6/22 21:01:16 2016/6/22 21:01:16
```

### 5. 编辑和打开文档——edit, open

使用 open 打开目标文档，用 edit 编辑文档。支持绝对路径和相对路径。

```
[donggu @ hello]# edit hhh

Please input the content of file. End up input with double click Enter.
dfajsoihwoiregnjsdnflkdsflkjdsflksdajfaohfsdnhjkfdhghgskjdfllkf

[donggu @ hello]# open hhh
dfajsoihwoiregnjsdnflkdsflkjdsflksdajfaohfsdnhjkfdhghgskjdfllkf
```

### 6. 删除和格式化——rm, rmall

使用 rm 删除目标文件/文件夹，用 rmall 进行格式化。

注意：不能删除当前目录或当前目录的父目录。必须离开当前目录才能进行删除。

```
[donggu @ /]# rm dir1
[donggu @ /]# ls
name          type    owner   size   createTime      lastUpdate
publicFile    <dir>   public  0       2016/6/22 2:32:25 2016/6/22 2:32:25
[donggu @ /]# cd publicFile/hello
[donggu @ hello]# rm publicFile
Error: Cannot remove father folder "publicFile", please jump to higher folder.
[donggu @ hello]#

[donggu @ hello]# rmall
[donggu @ /]# ls
name          type    owner   size   createTime      lastUpdate
[donggu @ /]#
```

## 7. 获取帮助

help 命令显示帮助。

```
[donggu @ /]# help
=====
                        欢迎使用文件管理系统
                        1452764 何冬怡
=====
本系统使用类Linux命令，支持下列命令
ls <path>          列出当前目录下所有文件/文件夹
cd <path>          跳转到目标路径，支持..回到上一级, 绝对路径及相对路径
mkdir <name>       在当前目录下新建文件夹
mkfile <name>      在当前目录下新建文档
rm <name>          删除当前目录下的文件/文件夹
edit <name>        编辑文件内容
open <name>        打开文件内容
rmall              系统格式化
help              获取帮助
quit              退出系统
=====
[donggu @ /]#
```

## 8. 退出系统

输入 quit 退出系统。显示成功退出后按 Enter 关闭窗口。文件系统信息将在退出时自动保存到 filesys 文件（若不通过 quit 退出则不保存）。

```
[donggu @ /]# quit
Quit successfully.
```

## 程序特性

### 9. 程序环境

使用了 C# 控制台完成程序。

### 10. 底层结构

利用 1024x1024 个 byte 模拟内存块组（Memory 类）。对应有 FAT 表由 1024 条 FAT 记录组成（FAT 类）。FAT 记录记录了物理内存的使用信息和连续情况，同时有 FCB 记录逻辑上的文件目录结构（FCB 类）。

使用 FileSystem 类实例对底层进行操作，对前端进行了完整的封装。

### 11. 前端交互

模仿了 Linux 系统的交互模式和命令，利用 Shell 类完成了前台交互的功能。通过和 FileSystem 的耦合完成与底层的交接。

### 12. 权限管理

系统利用用户名进行了简单的写操作权限管理（读无限制），原则为用户不能修改（指增删）其它用户创建的文件/文件夹。同时两个特殊用户：public 和 root。public 的文件/文件夹可被任意用户修改，而 root 可以修改所有用户的文件。

### 13. 报错管理

命名空间中定义枚举类 Error 对各种报错进行统一管理。设置 showError 函数实现对报错的格式化输出显示，基本涵盖了所有常见的操作错误。

```
public void showError(Error error, string name)
{
    string msg = "Error: ";
    switch(error)
    {
        case Error.NotExist: msg += ("Folder or file \"" + name + "\" not exist."); break;
        case Error.HasExist: msg += ("Folder or file \"" + name + "\" has exist."); break;
        case Error.CommandNotFound: msg += "Command not found."; break;
        case Error.FileNotFound: msg += ("File \"" + name + "\" not found."); break;
        case Error.PermissionDenied: msg += ("Permission denied. Please login as the owner of \"" + name + "\" to establish operation."); break;
        case Error.NotFolder: msg += ("\"" + name + "\" is not a folder."); break;
        case Error.NotFile: msg += ("\"" + name + "\" is not a file."); break;
        case Error.NoSpace: msg += ("insufficient memory."); break;
        case Error.RmFather: msg += ("Cannot remove father folder \"" + name + "\", please jump to higher folder."); break;
    }
    Console.WriteLine(msg);
}
```

### 14. 文件目录

FileSystem.exe

可执行文件

filesys

文件系统信息本地化文件

各种.cs                      源代码文件

FileSystem\_API 文档.chm    具体函数 API 文档

## 成员变量/函数接口

---

详见 [FileSystem\\_API 文档.chm](#)