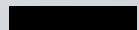


# 轻量级文件系统设计与实现



方梓畅、关宗杰、叶劲延

# content

## 目录

**01** 系统设计

**02** 实现细节

**03** 使用说明书

**04** 测试用例

**05** 团队分工

# 系统设计

---

01

# 架构概述

## 文件存储

所有数据统一存储在256MB的  
“light.fs” 文件中  
56MB用于系统信息，200MB用于数据  
存储  
每个存储单元为1MB

## 操作界面

支持命令行或图形界面操作  
提供初始化、创建、重命名、删除文件  
等基础功能

## 文件管理

显示文件内容、导入导出文件和存储统计信息  
支持灵活的存储布局，包括顺序文  
件排列、目录结构

## 存储系统特性

## 存储布局

支持类FAT链表结构、位图或索引节点  
方式  
能够处理文件重名问题

## 其他功能

提供文件的导入导出功能  
提供存储统计信息功能

# 算法说明

本次小组成员采用c++语言来完成文件系统的各种算法。

## 灵活接口设计

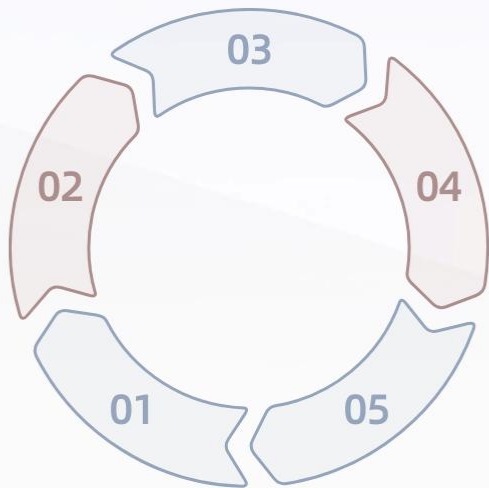
接口设计灵活，支持命令行和图形界面，适应多种使用场景。

## 数据存储单元

每1MB为一个单元，便于数据的高效管理和快速访问。

## 单文件策略

采用单文件'light.fs'策略，固定大小256MB，其中56MB用于系统信息，200MB用于数据存储。



## 多程序支持

团队可根据需求选择单个或多个可执行程序，满足不同场景需求。

## 核心功能覆盖

核心功能包括初始化、文件操作、导入导出等，确保高效的数据管理和用户体验。

# 实现细节

---

02

# 核心组件说明

LightFS 类核心实现解析



## 核心数据结构

LightFS 使用  
`std::vector<FileEntry>` 存储文件信息，确保高效访问与管理。



## 文件条目详情

每个 FileEntry 包含文件名、大小、内容及数据块索引，精细控制文件状态。



## 资源监控

`usedBlocks` 变量实时跟踪数据块使用情况，防止超出200MB用户数据限制。



## 初始化流程

构造函数创建256MB light.fs 文件，设置初始状态，构建自包含文件系统环境。

# 关键函数介绍

LightFS 类提供了多个关键函数来实现文件系统的功能需求



## 文件创建与删除

`createFile()` 创建新文件并分配唯一块索引，空间不足时提示用户。  
`deleteFile()` 删除文件，更新已用块计数。



## 文件列表与读写

`listFiles()` 展示所有文件及大小。`writeFile()` 和 `readFile()` 支持文件内容的写入与读取操作。



## 存储统计

`showStats()` 显示存储状态，包括已用和空闲空间，帮助监控文件系统健康。



## 互操作性增强

`importFile()` 和 `exportFile()` 实现文件导入导出，提升LightFS与外部环境的兼容性。



# 设计决策说明

## 设计决策说明

### 01

#### 单文件存储策略

采用统一的 light.fs 文件存储所有数据，简化管理与移植，提升系统整体效率。

### 02

#### 固定大小数据块

利用1MB固定大小数据块简化空间管理，通过 usedBlocks 追踪使用情况，避免复杂算法。

### 03

#### 内存中文件内容

文件内容驻留内存中的 FileEntry，牺牲持久化换取读写逻辑简化，增强系统易用性。

# 使用说明书

---

03

# 安装指南

代码适用于任何支持标准 C++ 的平台，包括 Windows、Linux 和 macOS。

编译环境与操作系统兼容性

## C++ 编译器需求

确保安装 GCC 或 Clang，支持 Visual Studio、Code::Blocks 或 g++ 编译。

## 跨平台兼容性

代码适用于 Windows、Linux 和 macOS，遵循标准 C++ 规范。

## 开发工具选择

灵活选择 IDE 或命令行工具，满足不同开发者偏好。本次实验使用命令行完成。

# 支持的文件格式/大小

01

## 文件最小大小

每个文件至少占用1MB，确保内容完整。

02

## 文件名长度限制

文件名最长255字符，超出部分将被截断。

03

## 总存储容量

总存储容量为256MB，包括元数据和用户数据。

04

## 元数据占用

56MB用于存储元数据，保证系统运行。

05

## 用户数据空间

200MB用于存储用户数据，满足基本需求。

06

## 最大文件数量

最多可存储200个文件，合理规划使用。

# 命令/操作指南



## 文件创建与删除

`createFile()` 创建新文件, 自动分配唯一块索引;  
`deleteFile()` 删除文件, 更新已用块计数。



## 文件列表与读写

`listFiles()` 展示所有文件及大小; `writeFile()` 和 `readFile()` 支持内容的写入与读取。



## 存储统计

`showStats()` 显示存储状态, 包括已用和空闲空间, 便于监控资源使用情况。



## 互操作性增强

`importFile()` 和 `exportFile()` 实现文件导入导出, 提升LightFS与外部环境的兼容性。

**测试结果**

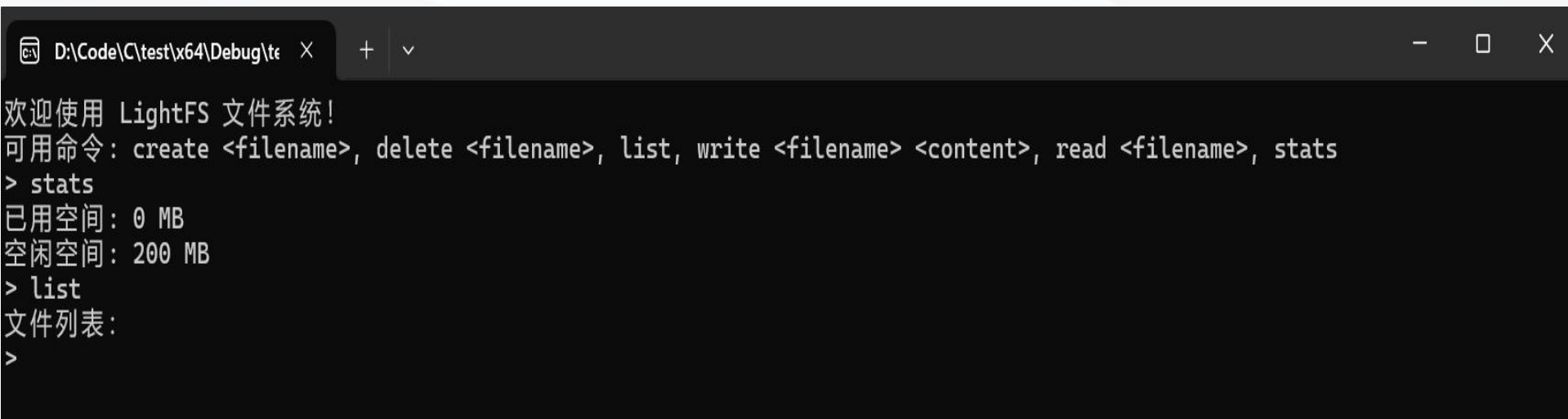
---

04

# 测试结果

我们实验小组用简单的命令行来完成文件系统的设计，接下来进行简单的测试，测试详细过程看录制视频。

## ①初始化



```
D:\Code\C\test\x64\Debug\te X + v
欢迎使用 LightFS 文件系统!
可用命令: create <filename>, delete <filename>, list, write <filename> <content>, read <filename>, stats
> stats
已用空间: 0 MB
空闲空间: 200 MB
> list
文件列表:
>
```

## ②创建文件

```
> create 操作系统.txt
文件 操作系统.txt 创建成功。
> stats
已用空间：1 MB
空闲空间：199 MB
> list
文件列表：
操作系统.txt (大小：0 字节)
>
```

## ③写入文件

```
> write 操作系统.txt 完成操作系统大作业
写入内容到文件 操作系统.txt 成功。
> stats
已用空间：1 MB
空闲空间：199 MB
> list
文件列表：
操作系统.txt (大小：18 字节)
>
```

## ④读取文件

```
> read 操作系统.txt
读取文件 操作系统.txt 的内容：[完成操作系统大作业]
```

## ⑤删除文件

```
> delete 操作系统.txt
文件 操作系统.txt 删除成功。
> list
文件列表：
> stats
已用空间：0 MB
空闲空间：200 MB
```



# 团队分工

---

05

# 工作分配



## 文档编写和视频录制

由叶劲延完成



## PPT制作

方梓畅完成主体框架设计，关宗杰进行  
PPT完善



## 代码开发

由组长关宗杰完成主体框架的设计，组员  
方梓畅进行代码查漏补缺，叶劲延进行测试用例的设计



# THANKS

