**目录**

[一、 引言 2](#_Toc19729)

[编写目的 2](#_Toc17427)

[项目背景 2](#_Toc16050)

[二、 项目概述 2](#_Toc9041)

[1．运行环境 2](#_Toc23981)

[2．条件与限制 2](#_Toc1252)

[三、 功能需求 2](#_Toc16558)

[1．功能划分 2](#_Toc9046)

[2. 功能描述 2](#_Toc27921)

[四、 其他需求](#_Toc17462) 3

[五、 系统基本结构设计 3](#_Toc17785)

[1. 系统设计 3](#_Toc18912)

[2. 磁盘结构. 3](#_Toc29856)

[3. 空白磁盘块管理 3](#_Toc15606)

[4. 系统流程描述 4](#_Toc15606)

[六、 数据结构设计 4](#_Toc655)

[1. 磁盘块结构 4](#_Toc18912)

[2. 超级块结构. 4](#_Toc29856)

[3. 文件信息结构 4](#_Toc15606)

[4. 目录项结构 5](#_Toc15606)

[七、 接口设计 5](#_Toc5130)

[用户接口 5](#_Toc10124)

# 一、引言

## 编写目的

用文件的形式，说明对微型文件系统的设计思路，包括该软件系统模块设计和划分、接口设计、数据结构、算法设计、变量和函数等的命名规范等，为程序的详细设计提供基础。

## 项目背景

项目名称：mini-FS

开发语言：C

# 二、项目概述

## 1．运行环境

软件运行环境：windows10操作系统。

## 2．条件与限制

在三周内完成软件的设计、开发、测试。统一使用C编程。

# 三、功能需求

## 1．功能划分

实现文件系统的管理功能，包括文件创建，查询，复制，删除，编辑，帮助等。

## 2. 功能描述

表一. 功能列表

|  |  |
| --- | --- |
| 函数名称 | 函数实现的功能 |
| create | 创建空间 |
| mount | 挂载空间 |
| fmt | 格式化当前空间 |
| dr | 显示文件目录：dir |
| cp | 复制文件：copy |
| dl | 彻底删除文件 |
| del | 删除文件（之后可以恢复） |
| map | 显示指定文件在空间所占用的全部块号 |
| opt | 优化空间，单个文件连续存放 |
| att | 显示文件属性 |
| help | 帮助信息 |
| rewrite | 编辑文件 |
| rst | 恢复回收站中的文件 |
| empty\_bin | 清空回收站 |
| add | 新建文件 |
| rename | 重命名文件 |
| tp | 显示文件内容 |
| close | 退出 mini-FS 返回操作系统 |

# 四、其他需求

1. 能够充分利用存储空间，存储效率比较高。

2. 删除文件的空间能够循环利用。

3. 算法实现不能太复杂，效率尽可能高，执行时间不能过长。

# 五、系统基本结构设计

## 系统设计：

我们将总信息分为3大类，超级块、文件控制块、以及文件信息快。1G的磁盘空间也将分为这三部分。同时以“块”作为存储的最小分割单位，一个块大小为4KB。

**1.1 超级块**

超级块用于记录磁盘信息和磁盘空间分配的信息。包括文件系统大小、块大小、空闲块的多少，以及空闲文件控制块链表的表头指针、存放了内容的文件控制块的表头指针以及空闲文件信息块的表头指针。

**1.2 文件控制块**

记录所有文件的文件名和文件信息，以及文件信息在磁盘块中存放的地址。

**1.3 文件信息块**

分块存储文件内容。存储文件时，优先寻找连续空间存储。

## 2. 磁盘结构

|  |  |  |
| --- | --- | --- |
| **超级块**  **（1MB）** | **文件控制块**  **（3MB）** | **文件信息块**  **（1020MB）** |

## 3. 数据结构设计

**超级块（super\_block）结构**，用来实现对空闲盘块使用情况的总体记录

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 作用 |
| Fs\_size | Int | 文件系统大小 |
| Block\_size | Int | 数据块的大小 |
| Free\_block\_size | Int | 空闲块的数量，即空闲空间的大小 |
| Free\_fcb | FCB \* | 指向第一个空闲文件控制块 |
| Full\_fcb | FCB \* | 指向第一个存放了内容的文件控制块 |
| Bitmap[] | Char | 用位图记录磁盘块的使用信息 |

**文件控制块（FCB）**，用来记录文件目录和文件信息

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 作用 |
| Name[9] | char | 文件名 |
| Exname[4] | char | 拓展名 |
| filetime | Time\_t | 文件创建时间 |
| filesize | int | 文件大小（长度） |
| Fcb\_nxt | Char \* | 指向下一个文件控制块的指针 |
| Filelocal[9] | Int | 指向文件内容占用的物理块编号 |

一个FCB块占64B大小，即一个4KB大小的块可以存放64个，本系统最多可存放3MB/4KB \* 64 = 49,152个文件。

采用二级磁盘块索引方式，最大可以存放8\*4KB+4K\*4KB/4约等于4MB大小的文件。

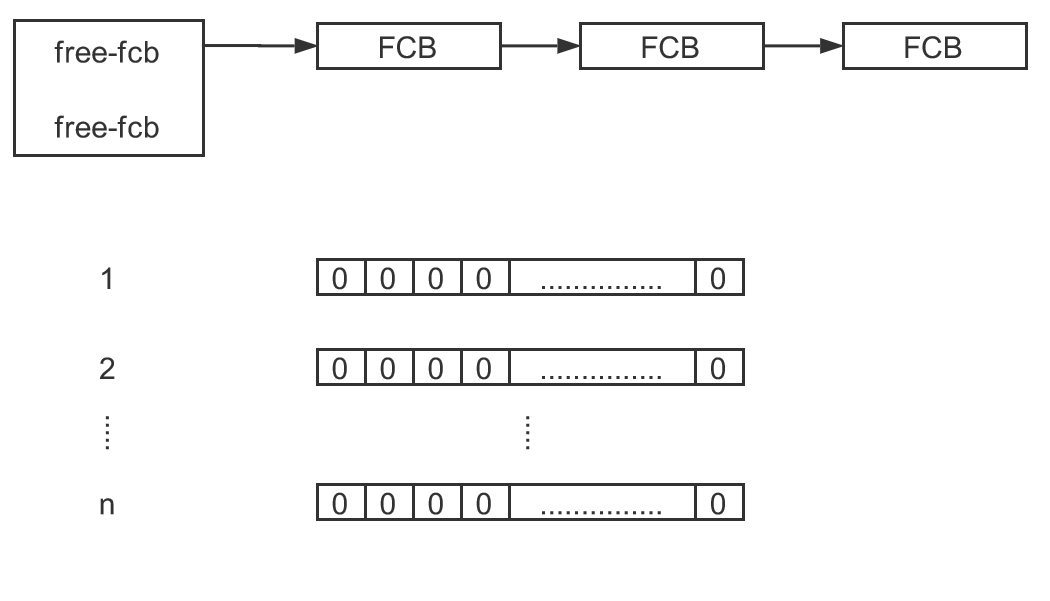
## 系统流程描述

* 1. **创建空间create**

通过create命令创建一个连续的大小为1G的文件，该文件即为模拟的mini-FS空间。

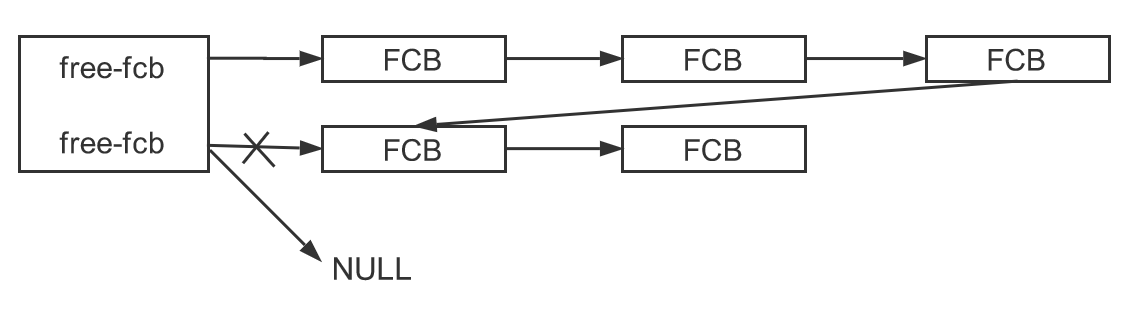
* 1. **开启空间mount**

通过mount命令安装空间，将超级块内容读入内存，之后超级块内容将一直留在内存中方便读写。同时输出命令提示，即提供给用户的指令和指令用法。



* 1. **格式化空间fmt**

通过fmt命令格式化空间，将free\_fcb的表尾指向full\_fcb的表头，相当于将所有的FCB块放入free\_fcb的链表中，之后让full\_fcb（非空闲文件控制块）指向null。将位示图中的每一位都置0，表明每一个磁盘块都是空闲的。



* 1. **帮助指令help**

从输入指令获取“help”时，直接打印mini-FS系统帮助信息。

* 1. **彻底删除文件dl**

从输入指令获取“dl”时，首先在文件控制块中根据文件名和扩展名查找该文件是否存在，若不存在，则返回error;若存在，则释放文件所存放的磁盘块，并将bitmap中的对应位置0。

* 1. **删除文件del**

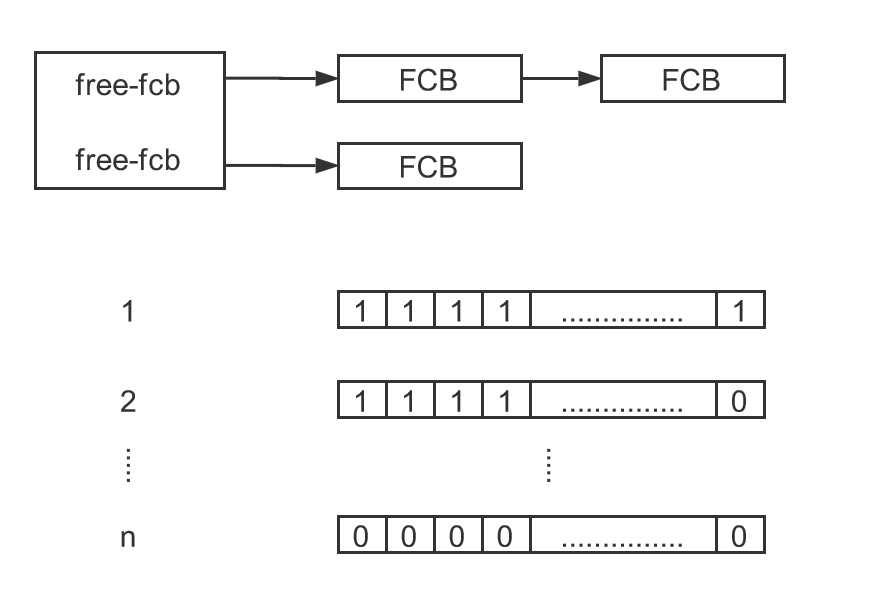
删除的时候FCB直接从full\_fcb的链表中移动到了free\_fcb的链表，其他不做处理。

* 1. **显示文件信息att**

从输入指令获取“att sFName”时，首先在文件控制块中根据文件名和扩展名查找该文件是否存在，若不存在，则返回error；若存在，则打印文件名、扩展名、文件创建日期和时间、文件大小等信息。

* 1. **复制文件cp**

从输入指令获取“cp”时，首先根据路径寻找进行复制的文件，如果没有找到则返回error；若存在，则先将文件的大小和磁盘剩余空间进行比较，如果小于磁盘剩余空间则输出空间不足，无法存放；如果大于磁盘剩余空间，则将一个空闲的FCB块移动到full\_fcb链表后，记录下文件信息，并将需要复制的文件一块一块复制进入空白文件块中，将位图中的对应磁盘块标记为已使用。



* 1. **显示文件所在块号map**

遍历full\_fcb的链表，找到需要显示的文件对应的FCB块。判断文件大小是否大于28KB，小于28KB则直接输出FCB块中filelocal数组（记录了文件所存的磁盘块号）的内容；大于28KB时，filelocal数组记录了索引块的块号，索引块里记录了文件所存的磁盘块号，输出这些块号。

* 1. **优化空间opt**

从后往前遍历文件块，如果有足够的文件空间，就与中间镶嵌的文件交换块的位子，如果没有，就拷贝到最后。

* 1. **显示目录dr**

输入dr命令后，申请一个FCB\*类型的指针dr\_temp，调用超级块中指向第一个存放了内容的文件控制块Full\_fcb初始化dr\_temp，进入循环，条件为dr\_temp为NULL时跳出循环，并打印出dr\_temp指向的FCB对象的成员：文件名name以及’.’与成员：扩展名exname。然后更改dr\_temp的值为当前dr\_temp指向的FCB对象的成员：指向下一个文件控制块的指针Fcb\_nxt，如此反复至达到循坏结束条件跳出后，结束dr命令进程。

* 1. **关闭系统close**

将内存中的超级块和目录信息都写入外存，关闭该文件。

* 1. **编辑文件rewrite**

搜索Full\_fcb链表，找到对应文件，输出文件大小，询问从哪个字节后开始修改。输入文本内容，将对应需要修改的地方覆盖，并视情况修改文件大小

* 1. **恢复文件rst**

把需要恢复的文件的FCB从free\_fcb的链表中挪动到full\_fcb的链表中。

* 1. **清空回收站empty\_bin**

具体操作是遍历free\_fcb（存有被删除的FCB）的链表，把每一个被删除文件所占用的空间都置0，并且把free\_fcb的链表清空。清空回收站之后被删除的文件将无法被恢复。

* 1. **新建文件add**

定义FCB类指针，指向Full\_fcb的末端，创建一个节点，并申请一块文件块连接上FCB，完成初始化，并设置文件大小为1，询问是否进行修改，若修改则调用修改函数。

* 1. **重命名rename**

首先在full\_fcb链表中寻找需要改名的文件是否存在，如果存在则在full\_fcb链表中寻找新的名字是否有重名文件。有重名文件则禁止重命名，没有重名文件则将该文件的FCB中的文件名和扩展名更改为新的名字。

* 1. **显示内容tp**

首先在full\_fcb链表中寻找要显示内容的文件是否存在，如果存在，文件大小小于28KB时，通过localfile[]数组里记录的磁盘块号一个个读取磁盘内容，将文件内容打印到命令行页面；如果文件大小大于28KB，则通过localfile[]数组读入索引块内容，通过索引块内容读入文件所存储的磁盘块号，读取磁盘内容，将内容打印到命令行页面。

## 系统空白块管理

在外存中建立一张位示图（bitmap），记录磁盘块的使用情况，0表示该磁盘块空闲，1表示该磁盘块占用。通过字节号和位号计算出相对于文件头的偏移量来对文件系统进行更改。

# 七、接口设计

## 用户接口

|  |  |  |
| --- | --- | --- |
| 创建系统 | create | 为mini-FS系统分配空间 |
| 启动系统 | mount | 安装mini-FS系统 |
| 格式化当前磁盘 | fmt | 调用fmt（）函数时格式化当前磁盘 |
| 显示文件目录 | dr | 显示系统中的文件目录 |
| 复制文件 | cp（f1，f2） | 实现将f2复制给f1 |
| 彻底删除文件 | dl（sFName） | 删除空间内的文件 |
| 删除文件 | del（sFName） | 删除空间内的文件，可在回收站中恢复 |
| 显示文件所在块号 | map(sFName) | 显示指定文本文件所占用的全部块号 |
| 优化空间 | opt | 优化空间，单个文件连续存放 |
| 显示文件属性 | att(sFName) | 显示文件属性，包括文件名、扩展名、文件创建日期和时间、文件大小等 |
| 帮助信息 | help | 输出提供的用户接口和指令 |
| 退出 | close | 退出空间，返回OS |
| 编辑文件 | rewrite(sFName) | 在文件指定位置后面编辑内容 |
| 恢复文件 | rst(sFName) | 从回收站中恢复指定文件 |
| 清空回收站 | empty\_bin | 把回收站中的文件全部删除，清空回收站之后被删除的文件将无法被恢复。 |
| 新建文件 | add（sFName） | 在mini-FS空间中新建文件 |
| 重命名 | rename(f1,f2) | 实现将f1重命名为f2 |
| 显示文件内容 | tp(sFName) | 在mini-FS空间中显示文件的内容 |