**操作系统课程设计实验报告**

实验题目： 简单文件系统的实现

姓 名：

学 号：

组 号：

专 业： 计算机科学与技术

班 级：

老师姓名：

日 期： 2019 年 12 月 01日

目 录

[一 题目介绍 1](#_Toc515881940)

[二 实验思路 1](#_Toc515881941)

[三 遇到问题及解决方法 2](#_Toc515881942)

[四 核心代码及实验结果展示 2](#_Toc515881943)

[五 个人实验改进与总结 4](#_Toc515881944)

[5.1 个人实验改进 4](#_Toc515881945)

[5.2 个人实验总结 4](#_Toc515881946)

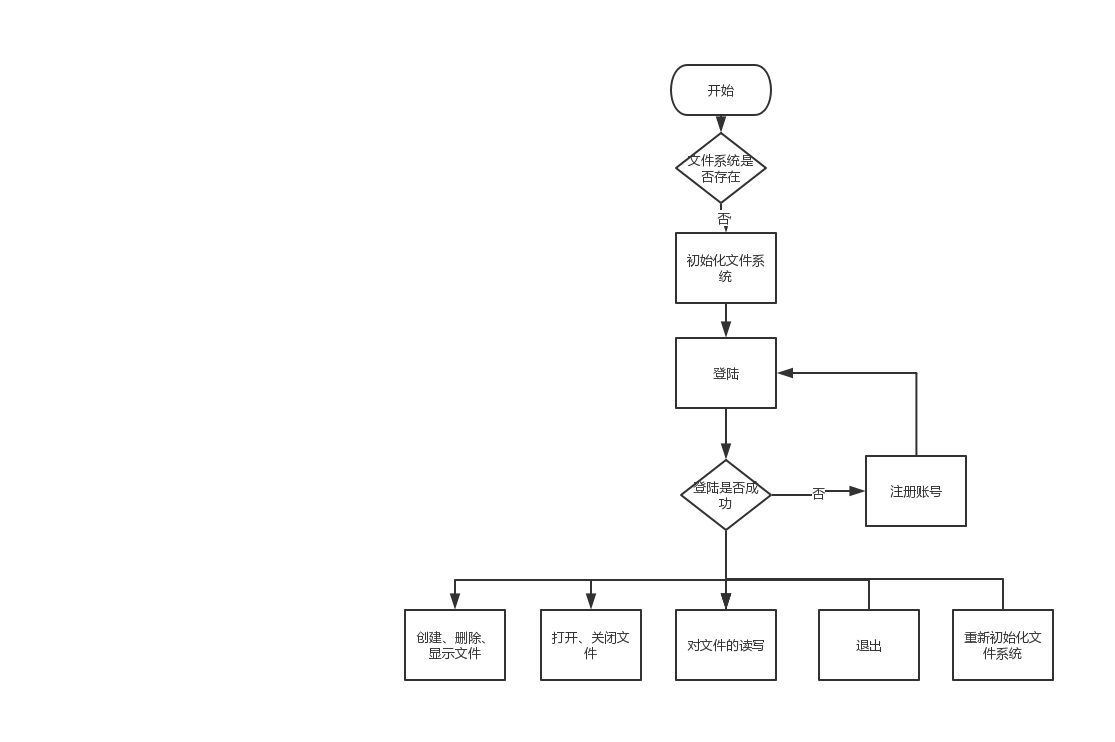
# 一 题目介绍

1. 在内存中开辟一个虚拟磁盘空间作为文件存储分区看，在其上实现一个简单的基于多级目录的单用户单任务系统中的文件系统，在退出该文件系统的使用时，应将虚拟磁盘上的内容以文件的方式保存到磁盘上，以便下次可以再将它恢复到内存的虚拟磁盘中
2. 要求提供以下操作命令：1.对文件存储器进行格式化，即按照文件系统的结构对虚拟磁盘空间进行布局，并在其上创建根目录以及用于管理文件存储空间等的数据结构 2.创建子目录 3.删除子目录 4.更改当前目录 5.创建文件 6.打开文件 7.关闭文件 8.显示目录中的内容 9.写文件 10 读文件 11删除文件 12 用于退出文件系统

# 二 实验思路

原理：设计的文件系统管理512个数据块每个数据块为512位。文件物理结构用顺序存储，空间磁盘管理用512位的位示图。用32个i节点来记录文件起始地址

总体思路（如图1）



具体函数、结构体解释解释：

typedef struct

{

char user\_name[10]; // 用户名

char password[10]; // 密码

} User;//用来存储用户信息

typedef struct

{

short inum; // 文件i节点号

char file\_name[10]; // 文件名

char type; // 文件类型 ‘-’ 文件 ‘d’文件夹

char user\_name[10]; // 文件所有者

short iparent; // 父目录的i节点号

short length; // 文件长度

//short address[2]; // 存放文件的地址

short start;//存放文件的起始地址

short blk\_num;//数据块数

} Inode;//存储i节点信息

typedef struct

{

short inum; // i节点号

char file\_name[10]; // 文件名

short mode; // 读写模式(1:read, 2:write,

// 3:read and write)

short offset; // 偏移量

} File\_table;//用来存储正在打开的文件表

void login(void);/\*用户登陆函数，若用户已存在则输入正确密码登入，若不存在，则建立新用户\*/

void init(void);/\*初始化文件系统，位示图全为0，所有i结点的父节点为0；并存入将信息存入本地\*/

int analyse(char \*);//分解用户命令，知道输入参数和具体命令

void save\_inode(int);//将i节点信息存入本地

int get\_blknum(int a);//申请连续a块空闲的磁盘块，返回初始磁盘块地址

void read\_blk(int b);//从磁盘块中读b个字节

void write\_blk(int num,int s\_start,int length)//从磁盘块中写入length长度的字节，num为初始磁盘块号，s\_start为开始写的位置\*/

void release\_blk(int num);//释放编号为num的磁盘块，num位置磁盘块设为0

void pathset();//更新当前路径

void delet(int innum);//删除目录树

int check(int i);//检查文件的权限是否属于该用户，如果返回1 ，属于

void help(void);//显示帮助文档

void cd(void)//切换目录

void dir(void);//显示当前目录下的子目录和文件

void mkdir(void);//在当前目录下创建子目录

void creat(void);//建立文件

void open(void);//打开文件 若文件不存在，提示

void read(void);//从文件中读取内容

void write(void);//向文件中写入内容

void close(void);//关闭文件

void del(void)//删除文件

void logout(void);//退出该用户

void command(void);//输入命令

void rd();//删除目录树

void quit();//退出系统

# 三 遇到问题及解决方法

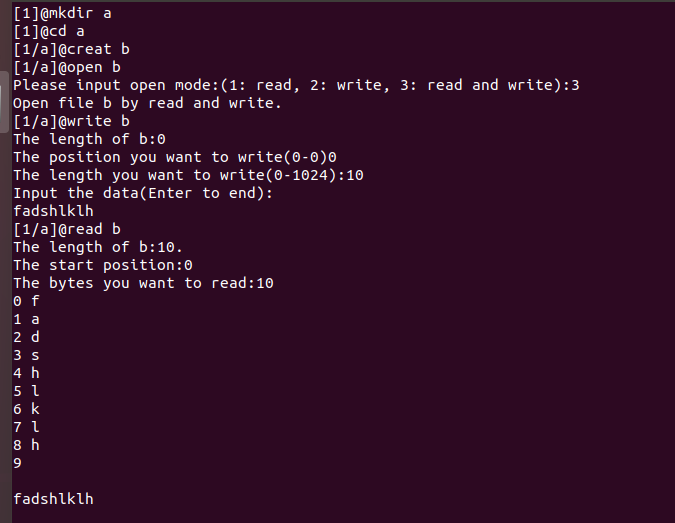
1. 本来只能申请一块磁盘块。后来根据申请连续空余n块的磁盘块，返回初始磁盘块号和磁盘块数量。
2. 写入磁盘块时，如果写入数量大于磁盘块剩余容量时，多余的数据不会读出。解决方法：

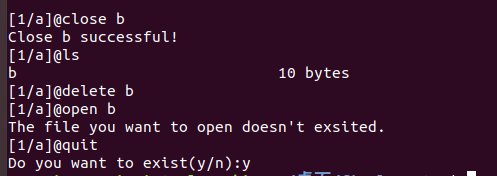
若大于原来的磁盘块，就查看该磁盘块后面的磁盘块是否空闲，若空闲，则申请。

3.写入磁盘块的时，无法从任意位置写入。解决方法：根据要写入的位置position，和输入的长度length，把原来文件中position后面的字节全部往后移length，然后再从Position位置写入数据

# 四 核心代码及实验结果展示

功能一结果：





# 五 个人实验改进与总结

1. 通过本次实验更深入的理解了文件系统的工作原理，对磁盘块、i节点、顺序存储、位示图有了更深入的理解
2. 实验改进想法：删除文件时，只能删除当前文件夹下的文件和空文件夹，若当前文件夹下仍有带有文件夹的文件夹，没有彻底删除，可以用递归的方式删除，文件夹下的所有文件及文件夹