实验报告

实验名称

简单文件系统的模拟

实验内容

模拟实现一个简单的文件管理系统，实现包括用户登录、文件创建、文件写入、文件读取、文件重命名、文件删除、查看磁盘状态等功能。

实验环境

- 编程语言：C
- 操作系统：Linux
- 编译器：gcc

实验思路

1. 设计合适的数据结构，用来实现二级目录。
2. 功能页面需要给出提示，随后进入用户页面。
3. 创建用户和登录用户。
4. 登录之后可以进行文件的具体操作，如权限修改，重命名等在内。
5. 可以退出用户，自由切换账号。
6. 输入exit后退出。

实验步骤

1. 数据结构设计

设计了以下主要数据结构：

- **磁盘块结构体**：用于管理磁盘的分配和释放。

```
typedef struct disk_block {
    int startpos;
    int useFlag;
    struct disk_block *next;
} diskNode;
```

- **文件表结构体**：用于管理文件的信息，包括文件名、起始位置、长度、最大长度、权限、创建时间等。

```
typedef struct fileTable {
    char filename[10];
    int startpos;
    int length;
    int maxlength;
    char fileKind[4];
    struct tm *timeinfo;
    struct fileTable *next;
    bool openFlag;
} fileTable;
```

- **用户目录结构体**：用于管理用户文件目录。

```
typedef struct userDirectory {
    struct fileTable *file;
    struct userDirectory *next;
} UFD;
```

- **主目录结构体**：用于管理用户信息。

```
typedef struct masterDirectory {
    char username[10];
    char password[10];
    UFD *user;
} MFD;
```

2. 磁盘初始化

初始化磁盘空间和用户表，并将磁盘的所有块标记为未使用。

```
void initDisk() {
    diskHead = (diskNode*)malloc(sizeof(diskNode));
    diskHead->startpos = 0;
    diskHead->useFlag = 0;
    diskHead->next = NULL;
    for(int i = 0; i < MAX_DISK; i++) {
        disk[i] = '0';
    }
    for(int i = 0; i < MAX_USER; i++) {
        userTable[i].user = NULL;
    }
}
```

3. 用户管理功能

实现了用户的创建、删除、登录和登出功能。

- **用户创建**：

```
void userCreate(char username[], char password[]) {
    if(userNum >= MAX_USER) {
        printf("The user table is full!\n");
        return;
    }
    for(int i = 0; i < MAX_USER; i++) {
        if(userTable[i].user != NULL) {
            if(strcmp(userTable[i].username, username) == 0) {
                printf("The user already exists!\n");
                return;
            }
        }
    }
    for(int i = 0; i < MAX_USER; i++) {
        if(userTable[i].user == NULL) {
            userTable[i].user = (UFD*)malloc(sizeof(UFD));
            userTable[i].user->file = NULL;
            userTable[i].user->next = NULL;
            strcpy(userTable[i].username, username);
            strcpy(userTable[i].password, password);
            userNum++;
            printf("User %s created successfully!\n", username);
            return;
        }
    }
}
```

- **用户登录**:

```
void userLogin(char username[], char password[]) {
    for(int i = 0; i < MAX_USER; i++) {
        if(strcmp(userTable[i].username, username) == 0) {
            if(strcmp(userTable[i].password, password) == 0) {
                printf("Login successfully!\n");
                currentUser = &userTable[i];
                return;
            } else {
                printf("Password error!\n");
                return;
            }
        }
    }
    printf("User not found!\n");
}
```

4. 文件管理功能

实现了文件的创建、写入、读取、重命名、删除等功能。

- **文件创建**:

```
void fileCreate(char fileName[], int length, char fileKind[]) {
    if(currentUser == NULL) {
        printf("Please login first!\n");
        return;
    }
    UFD *user = currentUser->user;
    fileTable *file = user->file;
    while(file != NULL) {
        if(strcmp(file->filename, fileName) == 0) {
            printf("The file already exists!\n");
            return;
        }
        file = file->next;
    }

    int startPos = requestDisk(length);
    if(startPos == -1) {
        printf("No enough space!\n");
        return;
    }

    fileTable *newFile = (fileTable*)malloc(sizeof(fileTable));
    strcpy(newFile->filename, fileName);
    newFile->startpos = startPos;
    newFile->maxlength = length;
    time_t t;
    time(&t);
    strcpy(newFile->fileKind, fileKind);
    newFile->timeinfo = localtime(&t);
    newFile->next = NULL;
    newFile->openFlag = false;
    if(user->file == NULL) {
        user->file = newFile;
    } else {
        file = user->file;
        while(file->next != NULL) {
            file = file->next;
        }
        file->next = newFile;
    }
    printf("File %s created successfully!\n", fileName);
}
```

- **文件写入**:

```
void fileWrite(char fileName[]) {
    if(currentUser == NULL) {
        printf("Please login first!\n");
        return;
    }
    UFD *user = currentUser->user;
    fileTable *file = user->file;
    while(file != NULL) {
        if(strcmp(file->filename, fileName) == 0) {
            if(file->openFlag == true) {
                printf("The file is already open!\n");
                return;
            }
            if(file->fileKind[1] != '1') {
                printf("You don't have the write authority!\n");
                return;
            }
            file->openFlag = true;
            printf("Please input the content:");
            char content[1000];
            scanf("%s", content);
            int length = strlen(content);
            if(length > file->maxlength) {
                printf("The content is too long!\n");
                file->openFlag = false;
                return;
            }
            for(int i = 0; i < length; i++) {
                disk[file->startpos + i] = content[i];
            }
            file->length = length;
            printf("Write successfully!\n");
            file->openFlag = false;
            return;
        }
        file = file->next;
    }
    printf("File not found!\n");
}
```

- **文件读取**：

```
void fileCat(char fileName[]) {
    if(currentUser == NULL) {
        printf("Please login first!\n");
        return;
    }
    UFD *user = currentUser->user;
    fileTable *file = user->file;
    while(file != NULL) {
        if(strcmp(file->filename, fileName) == 0) {
            if(file->openFlag == true) {
                printf("The file is already open!\n");
                return;
            }
            if(file->fileKind[0] != '1') {
                printf("You don't have the read authority!\n");
                return;
            }
            file->openFlag = true;
            for(int i = 0; i < file->length; i++) {
                printf("%c", disk[file->startpos + i]);
            }
            printf("\n");
            file->openFlag = false;
            return;
        }
        file = file->next;
    }
    printf("File not found!\n");
}
```

- **文件删除**：

```
void fileDelete(char fileName[]) {
    if(currentUser == NULL) {
        printf("Please login first!\n");
        return;
    }
    UFD *user = currentUser->user;
    fileTable *file = user->file;
    fileTable *preFile = NULL;
    while(file != NULL) {
        if(strcmp(file->filename, fileName) == 0) {
            if(file->openFlag == true) {
                printf("The file is open!\n");
                return;
            }
            file->openFlag = true;
            if(preFile == NULL) {
                user->file = file->next;
            } else {
                preFile->next = file->next;
            }
            freeDisk(file->startpos, file->maxlength);
            printf("Delete successfully!\n");
            file->openFlag = false;
            return;
        }
        preFile = file;
        file = file->next;
    }
    printf("File not found!\n");
}
```

5. 磁盘管理功能

实现了磁盘的分配和释放功能。

- **请求磁盘空间**:

```
int requestDisk(int length) {
    int blockNum = length / BLOCK_SIZE + (!!(length % BLOCK_SIZE));
    for(int i = 0; i < 1024; i++) {
        int pos = i;
        if(posTable[i] == 0) {
            int j = 0;
            for(j = 0; j < blockNum; j++) {
                if

(posTable[pos + j] != 0) break;
            }
            if(j == blockNum) {
                for(int k = 0; k < blockNum; k++) {
                    posTable[pos + k] = 1;
                }
                return pos;
            }
        }
    }
    return -1;
}
```

- **释放磁盘空间**:

```
void freeDisk(int startPos, int length) {
    int blockNum = length / BLOCK_SIZE + (!!(length % BLOCK_SIZE));
    for(int i = 0; i < blockNum; i++) {
        posTable[startPos + i] = 0;
    }
}
```

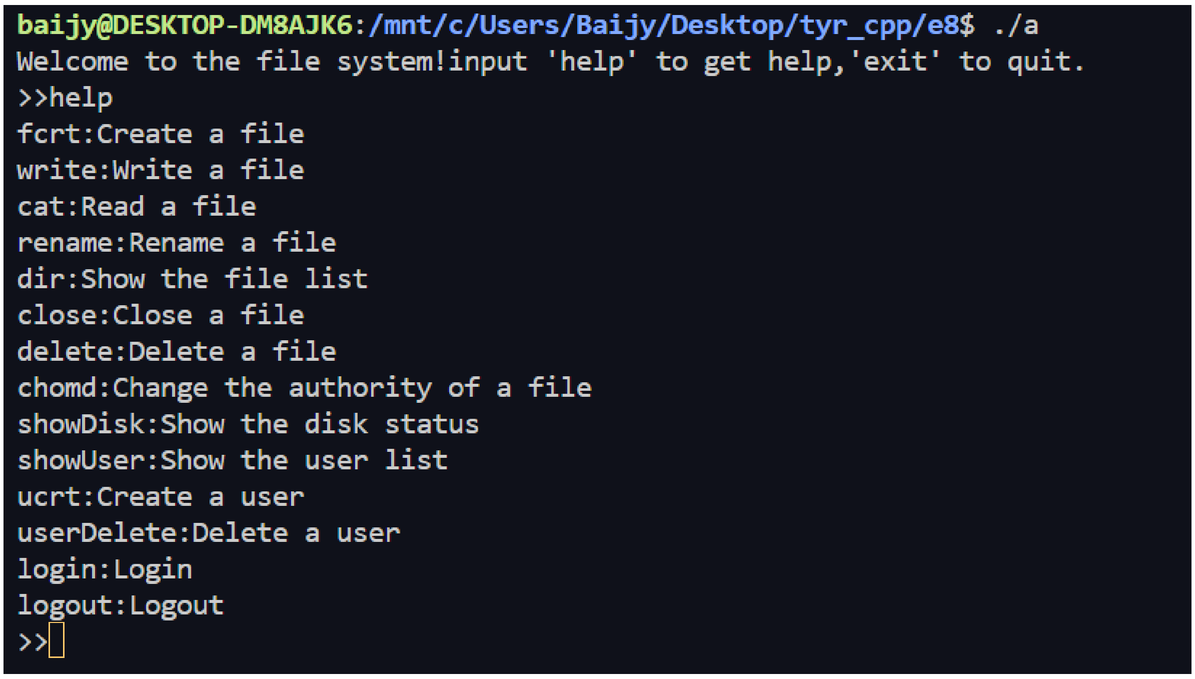
6. 时间管理功能

实现了文件创建和修改时间的记录功能。

```
fileTable *newFile = (fileTable*)malloc(sizeof(fileTable));
time_t t;
time(&t);
newFile->timeinfo = localtime(&t);
```

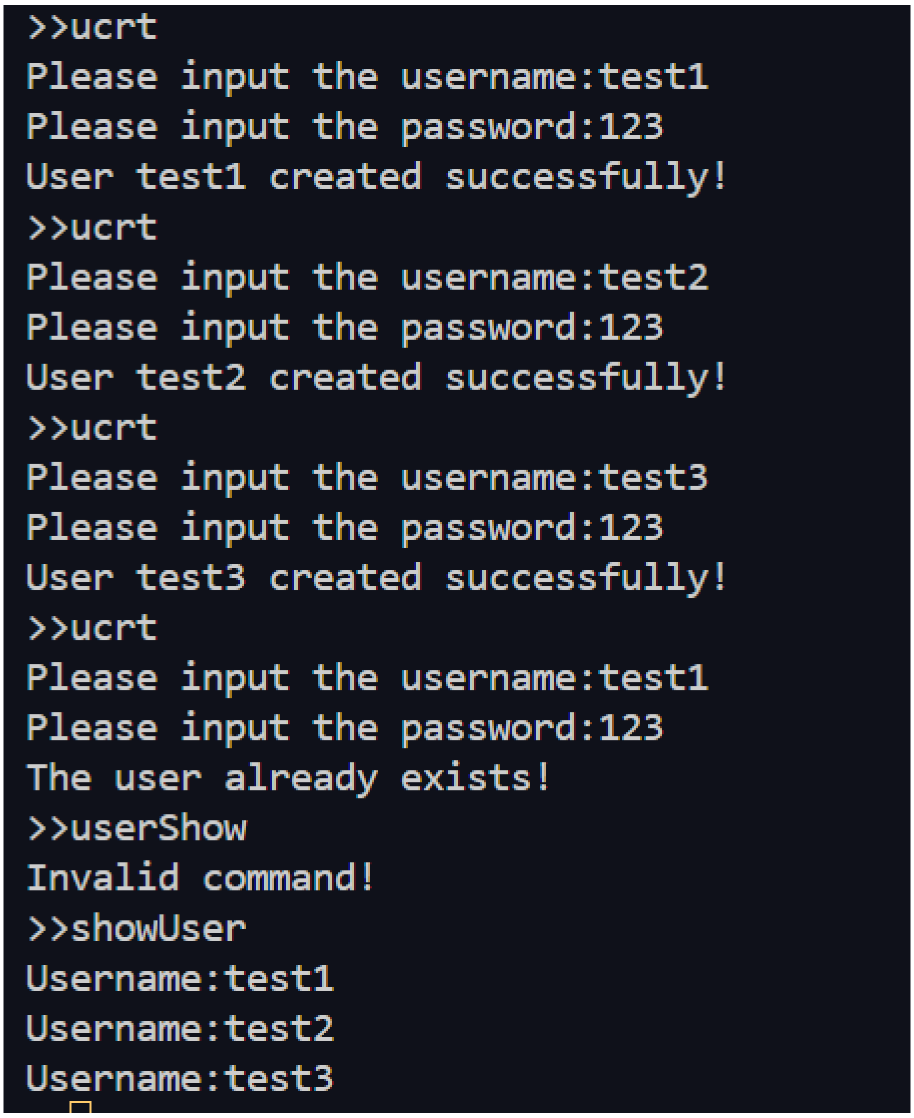
结果展示

help界面



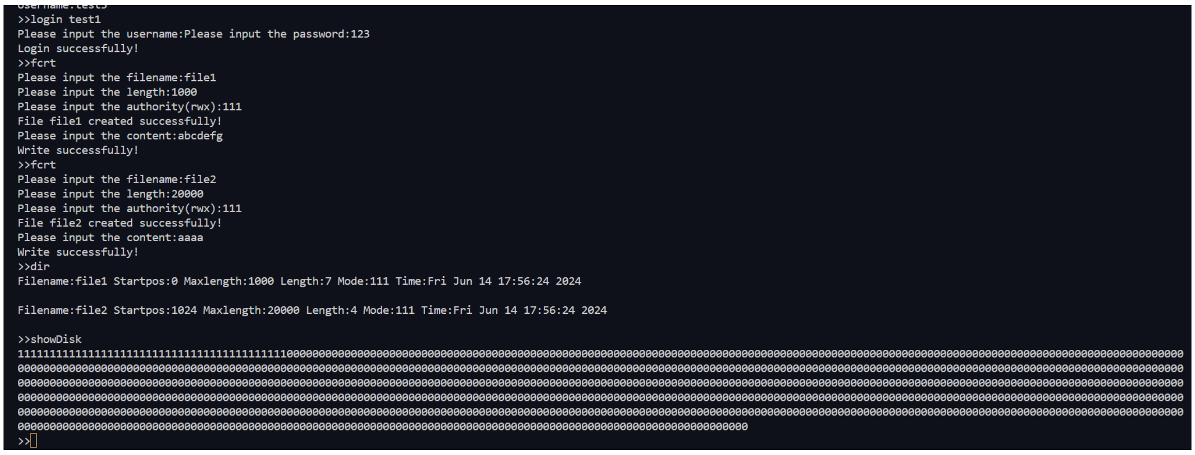
```
baijy@DESKTOP-DM8AJK6:/mnt/c/Users/Baijy/Desktop/tyr_cpp/e8$ ./a
Welcome to the file system!input 'help' to get help,'exit' to quit.
>>help
fcrt:Create a file
write:Write a file
cat:Read a file
rename:Rename a file
dir:Show the file list
close:Close a file
delete:Delete a file
chomd:Change the authority of a file
showDisk:Show the disk status
showUser:Show the user list
ucrt:Create a user
userDelete:Delete a user
login:Login
logout:Logout
>>
```

创建用户和展示当前用户



```
>>ucrt
Please input the username:test1
Please input the password:123
User test1 created successfully!
>>ucrt
Please input the username:test2
Please input the password:123
User test2 created successfully!
>>ucrt
Please input the username:test3
Please input the password:123
User test3 created successfully!
>>ucrt
Please input the username:test1
Please input the password:123
The user already exists!
>>userShow
Invalid command!
>>showUser
Username:test1
Username:test2
Username:test3
```

用户登录，新建文件，展示当前目录下的文件，展示磁盘使用情况

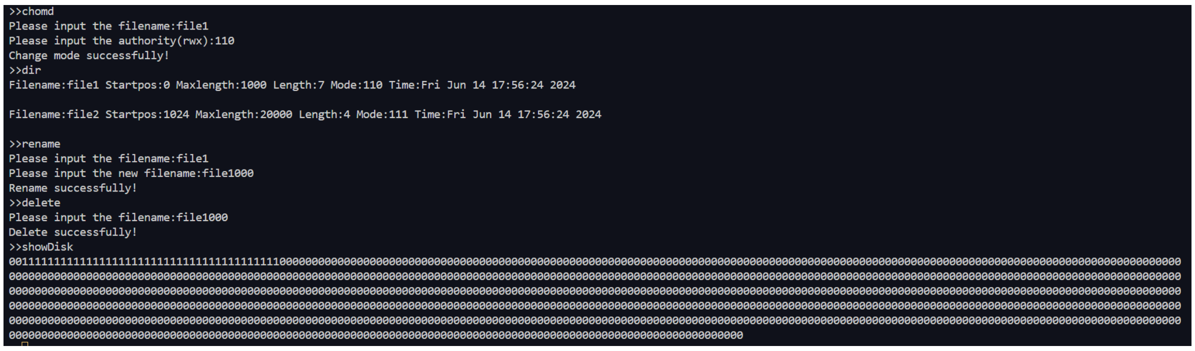


```
>>login test1
Please input the username:Please input the password:123
Login successfully!
>>fcrt
Please input the filename:file1
Please input the length:1000
Please input the authority(rwx):111
File file1 created successfully!
Please input the content:abcdefg
Write successfully!
>>fcrt
Please input the filename:file2
Please input the length:20000
Please input the authority(rwx):111
File file2 created successfully!
Please input the content:aaaa
Write successfully!
>>dir
Filename:file1 Startpos:0 Maxlength:1000 Length:7 Mode:111 Time:Fri Jun 14 17:56:24 2024

Filename:file2 Startpos:1024 Maxlength:20000 Length:4 Mode:111 Time:Fri Jun 14 17:56:24 2024

>>showDisk
11111111111111111111111111111111111111111100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
>>
```

文件权限修改，文件改名，删除文件



```
>>chomd
Please input the filename:file1
Please input the authority(rwx):110
Change mode successfully!
>>dir
Filename:file1 Startpos:0 Maxlength:1000 Length:7 Mode:110 Time:Fri Jun 14 17:56:24 2024

Filename:file2 Startpos:1024 Maxlength:20000 Length:4 Mode:111 Time:Fri Jun 14 17:56:24 2024

>>rename
Please input the filename:file1
Please input the new filename:file1000
Rename successfully!
>>delete
Please input the filename:file1000
Delete successfully!
>>showDisk
00111111111111111111111111111111111111111100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

总结

本次实验通过设计和实现一个简单的文件管理系统，加深了对文件系统各项功能和内部实现的理解，巩固了对C语言数据结构和算法的掌握，同时增强了对实际操作系统中文件系统管理机制的认识。比如其中的位图机制，如何串联起FCB等在内。