设计实现一个模拟文件系统

1.1 目的

1. 掌握操作系统的实现方法；
2. 设计并且实现一个模拟的文件系统。

1.2 内容

1. 基于一个 100M 的大文件模拟磁盘；
2. 格式化，建立文件系统管理数据结构；
3. 实现文件以及目录的创建和删除、目录显示等基本功能，可以扩充文件读写、用户登录、权限控制等其他功能。

1.3 设计

1.3.1 开发环境

笔记本电脑操作系统为 Windows 10 64 位系统，虚拟机软件位 Vmware Workstation，实验平台为 Linux Ubuntu 16.04，源内核版本为 Linux-4.15.0，图形编程库软件为 QT creator，处理器为 Intel Core i7-7500U@2.70GHz，内存为 2GB。

1.3.2 具体设计

磁盘空间采用连续分配方式，以 1 个 KB 的空间大小作为盘块的单位，采用位示图记录盘块分配的情况。文件目录采用树状目录，初始化根目录也就是初始的位置。整个磁盘的数据结构如图 1 所示。

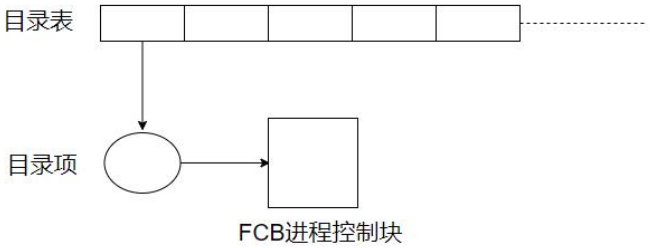

图 1 磁盘数据结构

目录表是存储目录项的数据机构。构造目录项，包括三个元素，分别是文件名，文件类型和文件控制块。对于文件控制块，通过结构体构造了记录一些信息的载体，记录的信息包括文件数据起始盘块号，文件的大小，数据的大小，读取数据的指针以及链接其他文件的数目。

构造的数据结构主要包括目录表，目录项，文件控制块，以及在实现多用户功能时构造的用户信息和全体用户记录结构体。如图 2，图 3 所示。

```
//目录项 64B
struct dirUnit{
    char fileName[59];  //文件名
    char type;  //文件类型,0目录，1文件
    int startBlock; //起始盘块
};
//目录表
struct dirTable {
    int dirUnitAmount;//目录项数目
    dirUnit dirs[dirTable_max_size];//目录项列表
};
//FCB
struct FCB {
    int blockNum;    //文件数据起始盘块号
    int fileSize;    //文件大小，盘块为单位
    int dataSize;    //已写入的内容大小，字节为单位
    int readptr;     //读指针，字节为单位
    int link;    //文件链接数
};
```

图 2 目录项、目录表和进程控制块的数据结构

```
//用户信息和全体用户
typedef struct usrinfo{
    char usrname[60];
    char usrpwd[60];
}usrinfo;

typedef struct sysusrs{
    int usrmap[100];//1时表示当前位置有用户占用，0时没有
    int usrnum;
    usrinfo infos[100];
}sysusrs;
```

图 3 实现多用户功能时构造的用户信息和全体用户记录结构体

设计好关键数据结构之后，要制定一些关键的操作函数。这里编写 main 程序、对磁盘进行整体操作的程序 DiskOperate.cpp 以及对具体文件进行操作的程序 FileOperate.cpp。在 main 函数中实现对各功能函数的调用，DiskOperation.cpp 中 initSystem 函数对系统进行初始化即创建空间；exitSystem 函数退出系统；getBlock 函数实现磁盘分配；releaseBlock 函数实现释放盘块。FileOperation.cpp 中 initRootDir 函数实现对根目录的初始化，showDir 展示当前目录，ChangeDir 切换目录，changeName 修改文件名或者目录名，creatFile 创建文件，creatDir 创建目录，creatFCB 创建进程控制块，addDirUnit 添加目录项，deleteFile 删除文件，releaseFile 释放文件内存，deleteDirUnit 删除目录项，deleteDir 删除目录，deleteFileInTable 删除文件或目录项，doRead 执行读操作，doWrite 执行写操作。

1.4 调试

1.4.1 步骤

完成 main.cpp、FileOperate.cpp 和 DiskOperate.cpp 的编写后，再编写 Makefile 统一对这些源程序文件进行编译，然后执行并进行相关测试。

1.4.2 具体调试

执行 Makefile，便可打印出自己设计的模拟文件系统界面，可以输入 root 密码进入 root 用户操作界面，也可以不输入密码或者输错密码进入普通用户操作界面，普通用户操作界面可以通过 sudo 命令通过再次输入 root 密码进入 root 用户操作界面，进而完成多用户登录功能，如图 4 所示。

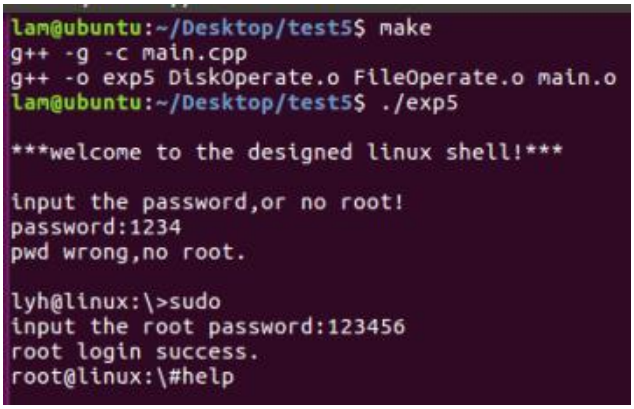

图 4 完成多用户登录操作

输入 help 命令可以打印出该模拟文件系统支持的所有命令，包括目录操作命令、文件操作命令和用户管理操作命令，如图 5 所示。

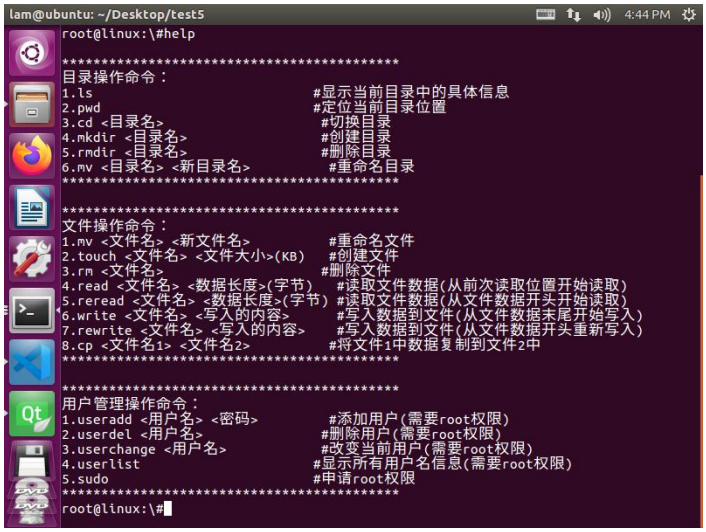

图 5 通过 help 命令打印出模拟文件系统支持的命令

对目录操作命令进行验证，通过 mkdir 命令创建目录，并通过 ls 命令查看当前目录下具体信息，通过 cd 命令切换目录，通过 rmdir 命令删除目录，再通过 pwd 命令定位当前目录位置，再通过 mv 命令重命名目录。如图 6 所示。

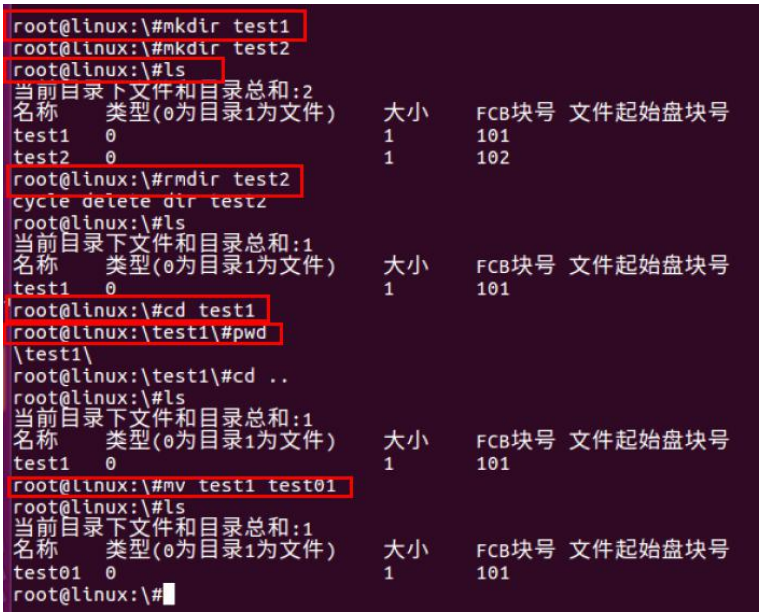

图 6 对目录操作命令进行验证

对文件操作命令进行验证，通过 touch 命令创建文件，通过 rm 文件删除文件，通过 mv 命令重命名文件，通过 write 写文件，通过 read 读文件，通过 rewrite 重写文件，通过 reread 重读文件。如图 7 所示。

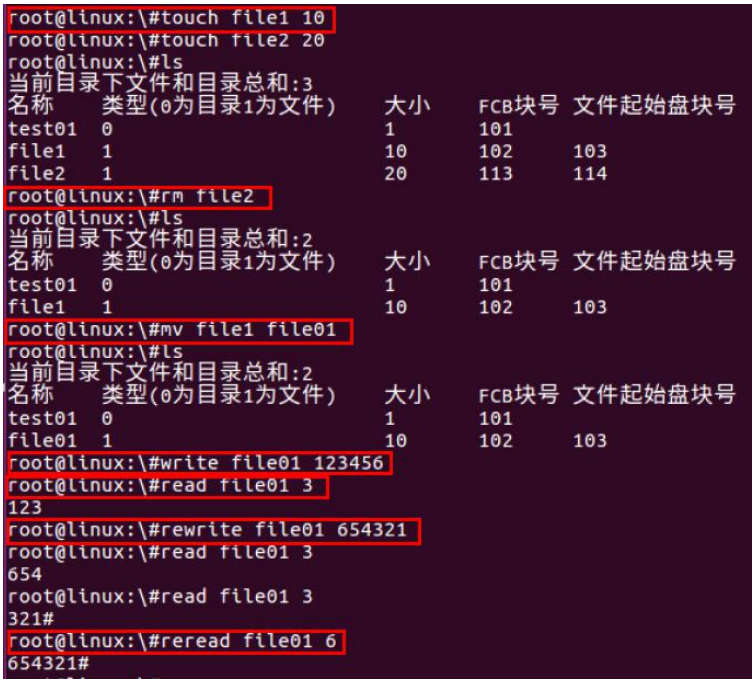

图 7 对文件操作命令进行验证

对用户管理操作命令进行验证，在 root 权限下通过 userlist 命令显示所有用户名信息，在 root 权限下通过 useradd 命令添加用户，在 root 权限下通过 userdel

命令删除用户，在 root 权限下通过 userchange 命令改变当前用户，普通用户通过 sudo 命令可以申请进入 root 用户操作界面。如图 8 所示。

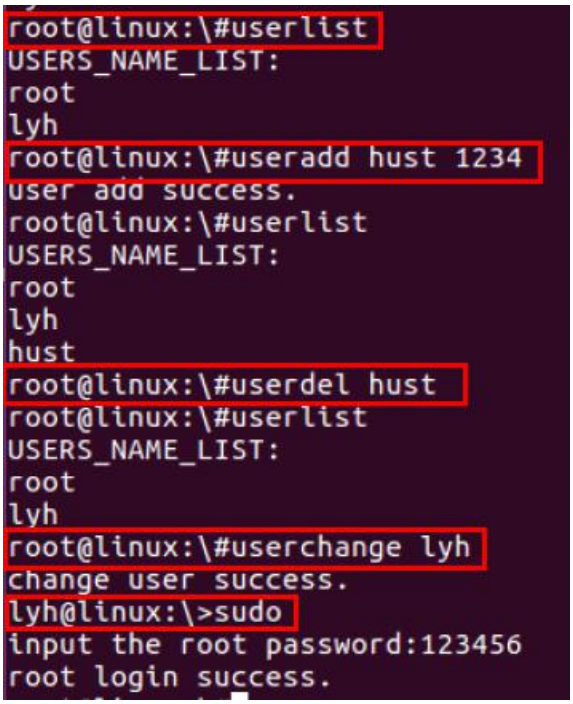

图 8 对用户管理操作命令进行验证