大连海事大学

OS 课程实践设计任务书

实验名称： Linux 文件系统的模拟实现

专业班级：计科三班

姓　　名：甘弘一

学　　号：2220180694

一、实验要求

1. 设计目的

(1)理解 Linux 文件系统的结构和管理机制（需先理解 Linux 文件系统的结构)。

(2)掌握 Linux 文件系统的多级目录管理、文件管理的实现方法。

2. 设计要求

(1)在理解 Linux 文件系统的数据结构的基础上，设计自己系统的数据结构和程序结构， 设计每个模块的处理流程，要求设计合理。

(2)编程序实现系统，要求实现可视化的运行界面，界面应清楚地反映出系统的运行结果；

(3)文件内容的读写都是对该文件数据块对应的一段内存空间内容的读写。真正的文件系统，在涉及到文件的读写、文件的创建等操作时，会用到内外存之间通信 的语句。使用内存来模拟外存的方式，利用内存的一段空间模拟 Linux 文件系统的实现。设 计结构体数组存放用户信息。设计文件和目录通用的结构体，用一个字段标识是文件或目录， 每个文件均有一个字段对应其所占用的数据块。对文件的内容的读写都是对该文件数据块对 应的一段内存空间内容的读写。

二、问题分析

需求分析(文件系统)

(1)有一个简单明了的界面。方便用户对功能的操作，以及文件系统的运用。

(2)可以在界面内进行输入命令，进而对程序命令的调用。大体上来说会设计。命令处理程序需实现创建文件、写文件、读文件、关闭文件、打开文件、创建目录、进入目录等基本命令功能。对文件系统的执行操作。

(3)可以对文件、目录、文件内容的具体信息的保存。通过磁盘文件，保存文件系统的内容，方便下一次查看，或者对数据的调用。

三、详细设计

3.1 初步设计(数据结构设计)

(1)Cd 命令：进入下一个子目录

大概构思：Curfcb 存放的是当前目录的信息，当进入下一级目录的时候，先遍历当前目录，查询是否与想进入的目录匹配，若是匹配，则修改当前目录，进入子目录。

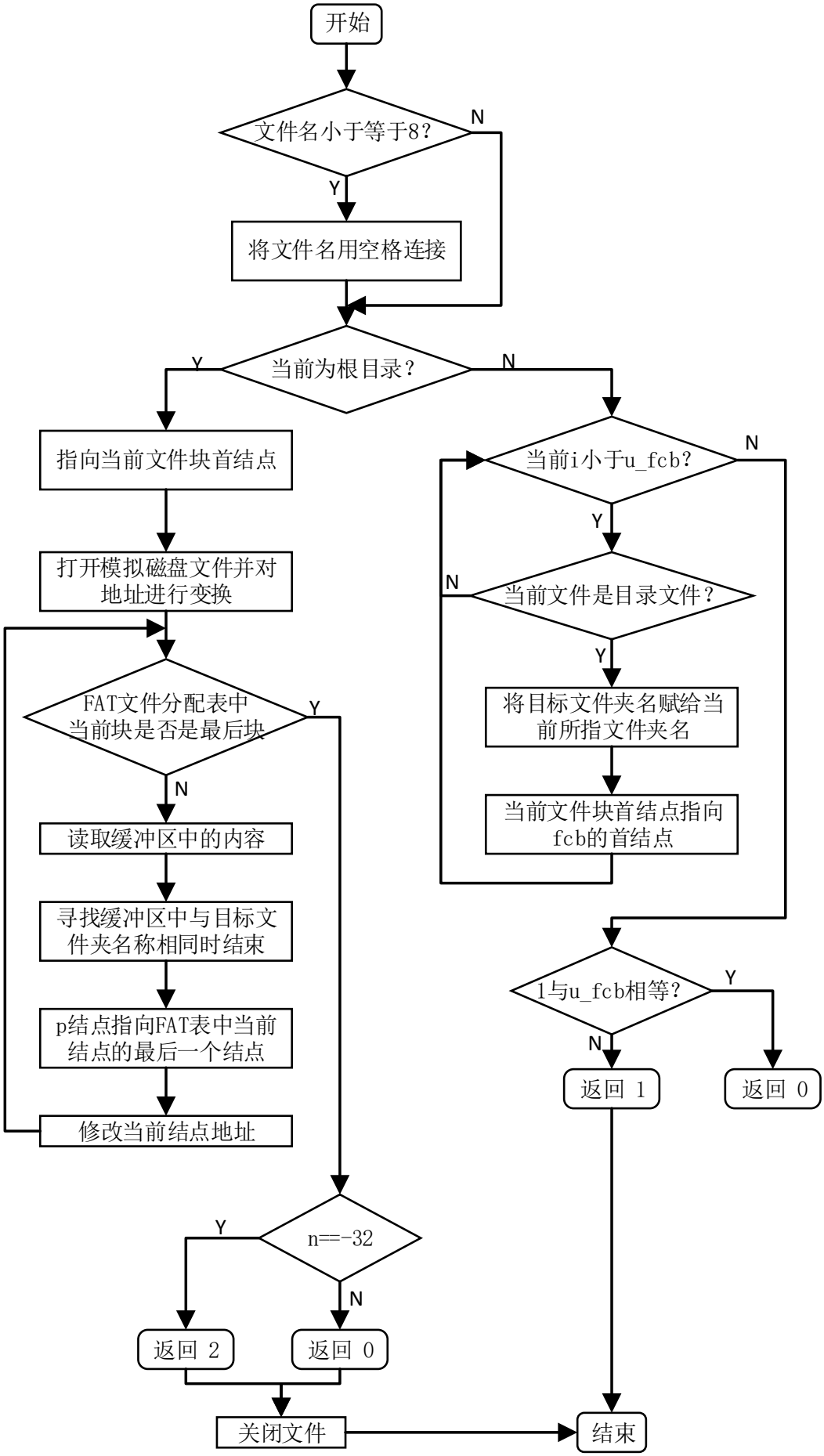
(2)输出 u_ofile 表

模拟 file 表的文件数据，用来保存数据方便查询。

(3)输出输出 open_file 表

查询打开的文件有多少个，方便对文件的读和写。

3.2cd_s 函数流程图设计


开始
文件名小于等于8?
N
Y
将文件名用空格连接
当前为根目录?
Y
N
指向当前文件块首结点
打开模拟磁盘文件并对地址进行变换
FAT文件分配表中当前块是否是最后块
Y
N
读取缓冲区中的内容
寻找缓冲区中与目标文件夹名称相同时结束
p结点指向FAT表中当前结点的最后一个结点
修改当前结点地址
当前i小于u_fcb?
N
Y
当前文件是目录文件?
N
Y
将目标文件夹名赋给当前所指文件夹名
当前文件块首结点指向fcb的首结点
1与u_fcb相等?
Y
N
返回 1
返回 0
n==-32
Y
N
返回 2
返回 0
关闭文件
结束

3.3 程序代码

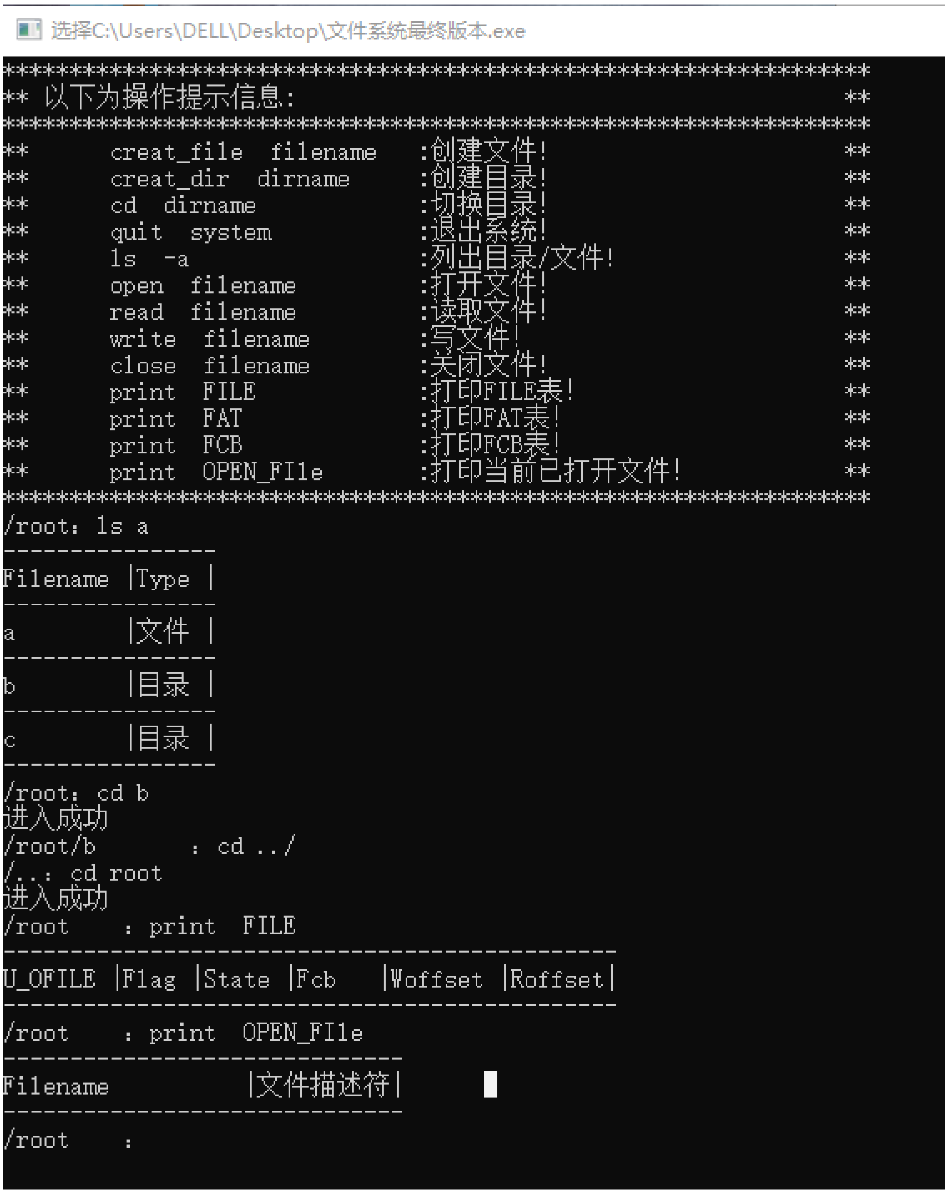
systate 为 0 表示回到根目录为 1 表示可以继续创建子目录和文件

```
int cd(char *dirname){
    char abuf[33]="0";
    int p,size,n,i,j,address,num=32;

    if(strlen(dirname)<=8){   //文件/目录名固定 8 位
        for(i=strlen(dirname);i<8;i++)
            strcat(dirname," ");
    }

    if(systate){//当前为根目录下
        p=curfcb.firstnode;
        size=curfcb.size;
        fd=fopen("模拟磁盘文件.txt", "r");
        address=(10+p)*BLOCKSIZE;//找根目录的数据区
        fseek(fd,address,SEEK_SET);
        n=size/32;//得到根目录下的 fcb 信息共 n 个 fcb

        while(n>0||FATTABLE[p].nextbl){
            if(n<32) num=n;
            for(i=0;i<num;i++){
                fread(abuf,sizeof(char),32,fd);
                if(abuf[9]=='1') continue; //a[9]=0 为目录，1 为文件
                for(j=0;j<8;j++)
                    if(abuf[j]!=dirname[j])//判断 cd 的目录名是否相同
                        break;
                if (j==8){//目录名存在
                    Setcurfcb(abuf);   //设置新的当前 curfcb
                    curfcb.index=-1;//设置为子目录
                    curfcb.address=(int)ftell(fd)-32; //保存当前子目录的 fcb
信息（根目录的数据区中）
//C 库函数 long int ftell(FILE *stream) 返回给定流 stream 的当前文件位置。
                    n=0;
                    break;
                }
            }
            //n-=32;
             p=FATTABLE[p].nextbl;//读取下一个物理块
            address=(11+p)*BLOCKSIZE;
            fseek(fd,address,SEEK_SET);
        }
        if(n==0) return 2;  //目录文件名字存在
```

```
        else return 0;
        fclose(fd);
    }

    else {  //当前目录为/..
        for(i=0;i<Nfcb;i++){
            if(fcb[i].type=='0'&&strcmp(fcb[i].filename,dirname)==0){
                strcpy(curfcb.filename,dirname);
                curfcb.firstnode=fcb[i].firstnode;
                curfcb.size=fcb[i].size;
                curfcb.index=i;
                break;
            }
        }
        if (i==Nfcb) return 0;
        else {
            systate=1;
            return 1;
        }
    }
}

void Setcurfcb(char *abuf){
    int j;
    char abuf2[8]="0";
    for(j=0;j<8;j++)
        curfcb.filename[j]=abuf[j];
    curfcb.state=abuf[8];
    curfcb.type=abuf[9];
    for(j=0;j<6;j++)
        abuf2[j]=abuf[10+j];
    curfcb.size= atoi(abuf2);
    for(j=0;j<3;j++)
        abuf2[j]=abuf[16+j];
    curfcb.firstnode= atoi(abuf2);
}

void root(){
    if(systate){
        systate=0;
        strcpy(curpath,"/..");
    }
}
```

```
void print_u_ofile()
{
//  system("color 5");
    printf("-----------------------------------------\n");
    printf("U_OFILE |Flag |State |Fcb   |Woffset |Roffset| \n");
    for(int i=0; i<10; i++)
    {
        if(file[i].f_state==-1)
            break;
        char flag[4];
        if(file[i].f_flag==0)
            strcpy(flag,"读");
        else
            strcpy(flag,"读&写");
        printf("-------------------------------------------\n");
        printf("%-8d|%-5s|%-6s|%-6d|%-8d|%-7d|\n",u_ofile[i],flag,"  使  用
",file[i].f_fcb,file[i].f_woffset,file[i].f_roffset);
    }
    printf("-----------------------------------------\n");
    return ;
}

void print_open_file()
{
    printf("----------------------------\n");
    printf("Filename          |文件描述符|\n");
    for(int i=0;i<100;i++)
    {
        if( strcmp(open_file[i].filename,"0")==0)
            break;
        printf("----------------------------\n");


    printf("%-18s|%-10d|\n",open_file[i].filename,open_file[i].file_fd);
    }
    printf("----------------------------\n");
}
```

四、实验结果



```
选择C:\Users\DELL\Desktop\文件系统最终版本.exe
****************************************************************
** 以下为操作提示信息:                                          **
****************************************************************
**      creat_file  filename    :创建文件!                     **
**      creat_dir  dirname      :创建目录!                     **
**      cd  dirname             :切换目录!                     **
**      quit  system            :退出系统!                     **
**      ls  -a                  :列出目录/文件!                **
**      open  filename          :打开文件!                     **
**      read  filename          :读取文件!                     **
**      write  filename         :写文件!                       **
**      close  filename         :关闭文件!                     **
**      print  FILE             :打印FILE表!                   **
**      print  FAT              :打印FAT表!                    **
**      print  FCB              :打印FCB表!                    **
**      print  OPEN_FIle        :打印当前已打开文件!           **
****************************************************************
/root: ls a
----------------
Filename  |Type |
----------------
a         |文件 |
----------------
b         |目录 |
----------------
c         |目录 |
----------------
/root: cd b
进入成功
/root/b        : cd ../
/..: cd root
进入成功
/root     : print  FILE
----------------------------------------------
U_OFILE  |Flag  |State  |Fcb    |Woffset  |Roffset|
----------------------------------------------
/root     : print  OPEN_FIle
------------------------------
Filename            |文件描述符|
------------------------------
/root     :
```

五、重点难点分析体会

(1)重难点分析

由于我自己的部分相对来说简单。首先得明白磁盘存储的构造（存储文件或目录数据为 32 位）。1-8 是对文件名或者目录名的保存。9 是可否被用。10 是用来区分是文件还是目录。11 到 16 是用来保存文件或目录数据的大小的。17-19 是盘块存文件的首地址。并且能通过计算得出其他需要的值。其次 cd 得通过输入的已有目录来对其进行处理。所以得先查找当前目录下是否有和输入值的目录。如果有就进行对字目录的读取，并且查找文件控制段转移位置到子目录所在地址。最后，再设置一个命令使得文件可以转换回根目录。

其实过程还挺简单的，只需要对 fcb 中存的数据的调用，就可以完成了。

也看了一下其段的代码，最难的部分还是对磁盘的编写。得需要在网上查相对应的资料才可以，明白原理。

(2)总结

在这将近两个星期的学习中，加强了我的操作系统的基础知识。明白了许多以往觉得缥缈的知识。对其他代码的研读，让我觉得原来编写一个这样的程序，虽然难，但特别有成就感。对于小组合作非常大愉快，小组讨论氛围浓厚。使得我们可以巩固学习的知识。