

实验 四 题目 文件系统 第 周星期

一、实验目的

模拟文件系统实现的基本功能,了解文件系统的基本结构和文件的各种管理方法,加深理解文件系统的内部功能及内部实现。通过用高级语言编写和调试一个简单的文件系统,模拟文件管理的工作过程,从而对各种文件操作命令的实质内容和执行过程有比较深入的了解。

二、实验内容和要求

编程模拟一个简单的文件系统,实现文件系统的管理和控制功能。要求本文件系统采用两级目录,即设置主文件目录[MFD]和用户文件目录[UED]。另外,为打开文件设置运行文件目录[AFD]。设计一个 10 个用户的文件系统,每次用户可保存 10 个文件,一次运行用户可以打开 5 个文件,并对文件必须设置保护措施。在用户程序中通过使用文件系统提供的 Create、open、read、write、close、delete 等文件命令,对文件进行操作。

三、实验主要仪器设备和材料

实验环境

硬件环境: 个人台式机 Microsoft Windows 7

软件环境: C 语言编程环境, vs 2010

四、实验设计方案及原理

1、实验原理

因为系统小,对文件目录检索使用简单的线性搜索,为了便于实现,对文件读写做了简化,在执行读写命令时,只修改读写指针,并不实际对文件进行操作,其中初始状态下,读指针=0,写指针=文件长度,文件保护码使用三位保护码: 1: 表示文件允许的读 2: 表示文件可写 3. 表示文件可执行,创建文件时通过模拟磁盘的使用情况来实现。

2、设计方案

程序采用两级目录结构,第一级设置主文件目录[MFD],第二级为用户文件目录[UFD],另外为打开文件设置运行目录[AFD]

```
struct mdf{                                /* 第一级: 主目录 MDF */
    char name[10];                          /* 用户名 */
    UFD* directory;                        /* 文件目录指针 */
    }maindir[10];                          /* 用户数组 */
typedef struct mdf MDF;

struct ufd{                                /* 第二级: 用户文件目录 UFD */
    char filename[10];                    /* 文件名 */
```

```

char procode[3];      /* 保护码 1:读 2:写 3:执行*/
int length;           /* 文件长度 */
}*p;

typedef struct ufd UFD;

struct afd{           /* 文件运行目录 */
char filename[10];    /* 文件名 */
char procode[3];      /* 保护码 1:读 2:写 3:执行*/
int rw;               /* 读写指针 初始状态读:0 写:文件长度 */
}afd[5];

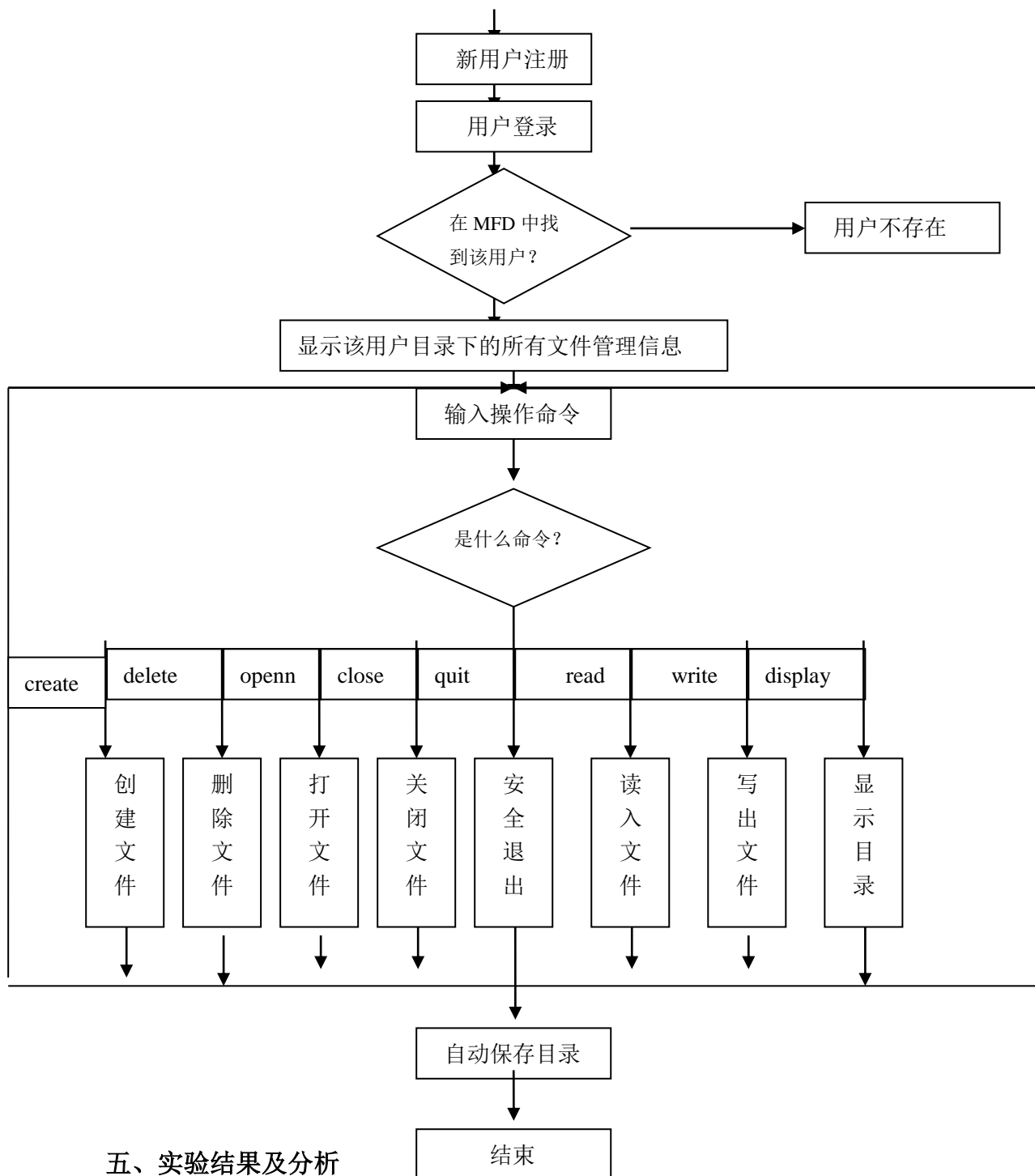
```

系统提供 8 条命令：create delete open close read write display quit，可以提供创建、删除、打开、关闭、读入、写出、显示、退出等功能。

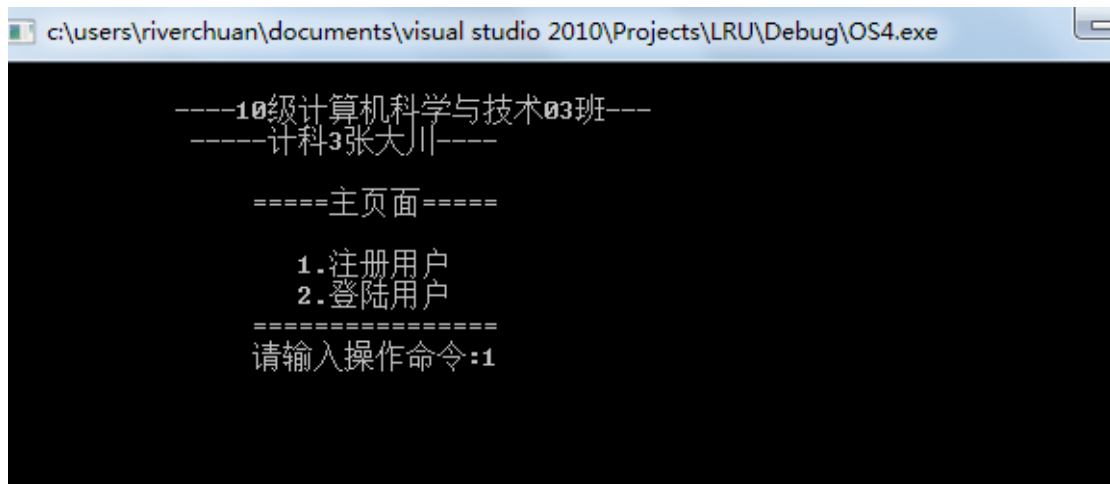
- ①、Create—在用户当前目录下创建一个文件，该文件的管理信息登录到用户文件信息管理模块中。
命令格式：create
- ②、delete—删除当前用户目录下的一个文件，删除后文件无法打开。
命令格式：delete
- ③、open—在当前用户目录打开某个文件并修改标志位，对文件各种操作都需先打开。
命令格式：open
- ④、close—关闭用户运行的文件。执行该命令后，用户在系统中运行文件状态位被修改为关闭。
命令格式：close
- ⑤、read—从用户已打开文件读信息，将文件信息读入并修改读取标志位，未打开文件无法读取。
命令格式：read
- ⑥、write—向用户已打开文件写入信息，修改标志位，未打开文件无法写出操作。
命令格式：write
- ⑦、display 用户目录文件列表显示。
命令格式：display
- ⑧、quit 用户注销命令。当使用该命令时，用户退出系统。
命令格式：quit

1、程序流程图：

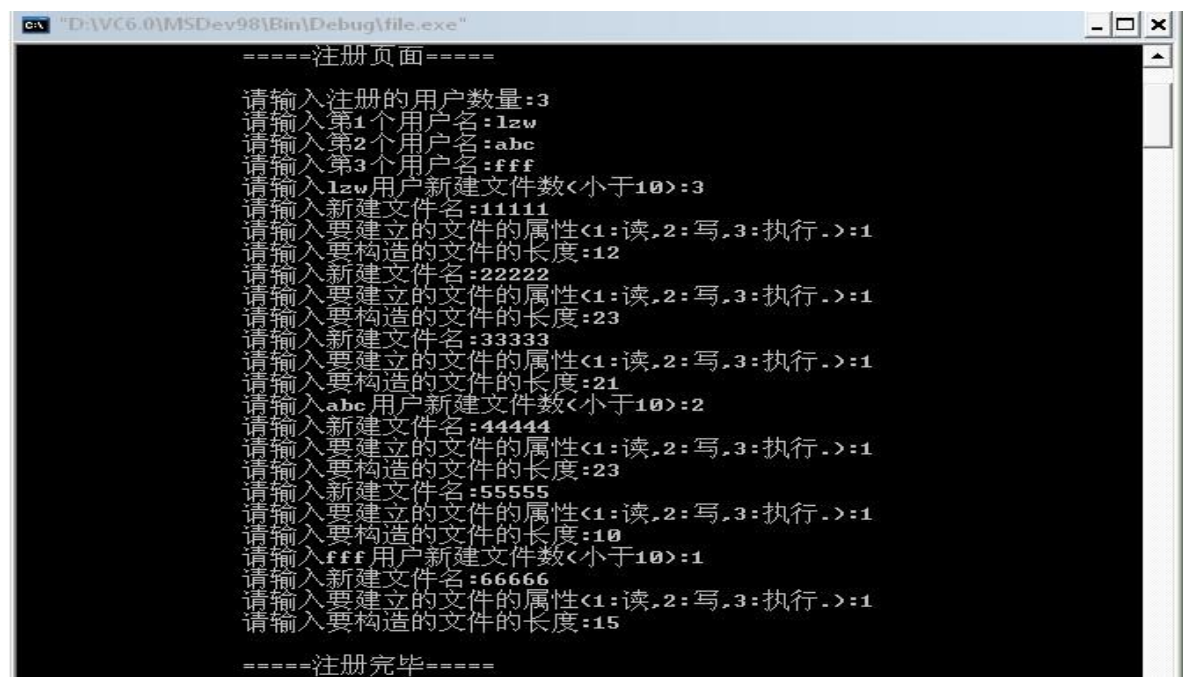
开始



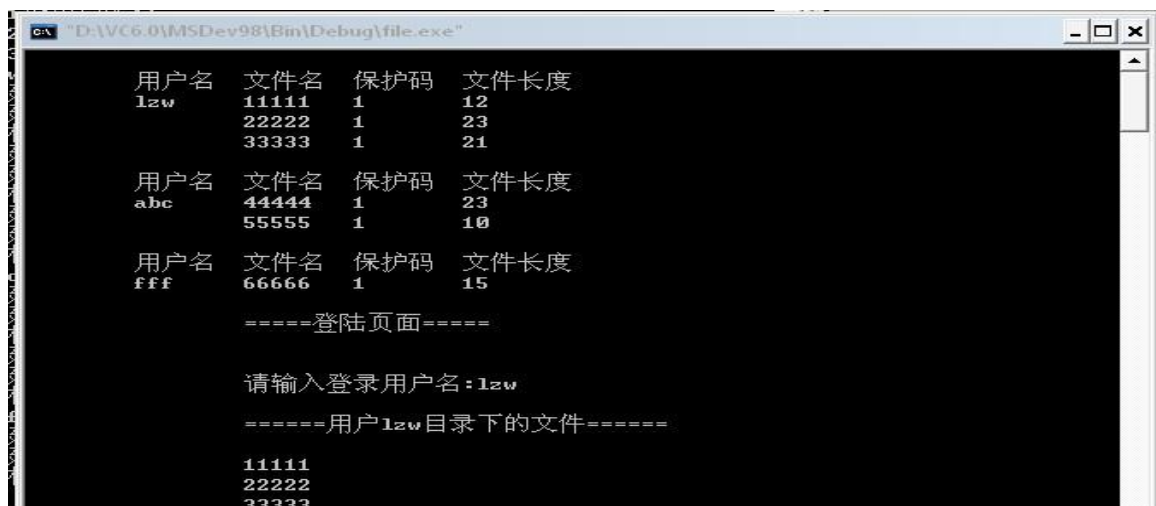
文件系统初始界面，第一次运行必须注册用户，注册用户的信息将会被自动写到默认目录的 database.txt 文件，如果文件存在可以直接登陆不用注册，注册将会覆盖原来的数据：



注册信息演示，用户数量最大为 10:



注册结束会自动转到登陆界面，而所有用户和目录文件信息将会显示在登陆界面上方，可以选择登陆一个用户：



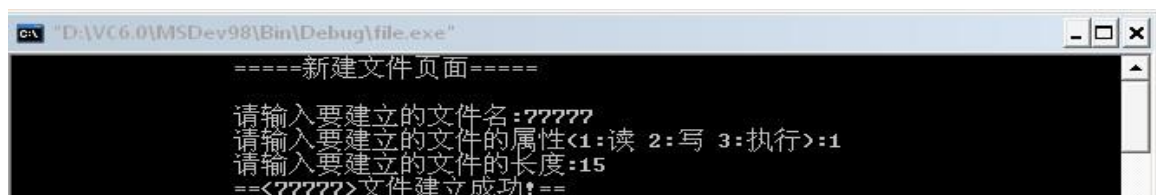
登陆成功则跳转至文件系统，可以输入相应命令对文件操作，这里测试先选择 create 命令新建文件



```
<<文件系统演示>>

*****
create 新建文件.
delete 删除文件.
open   打开文件.
close  关闭文件.
read   读出文件.
write  写进文件.
display 显示用户列表.
quit   安全退出.
*****
操作命令:create delete open close read write display quit
请输入操作命令:create
```

新建文件页面，按要求输入名字、属性、长度等：



```
=====新建文件页面=====

请输入要建立的文件名:77777
请输入要建立的文件的属性<1:读 2:写 3:执行>:1
请输入要建立的文件的长度:15
==<77777>文件建立成功!==
```

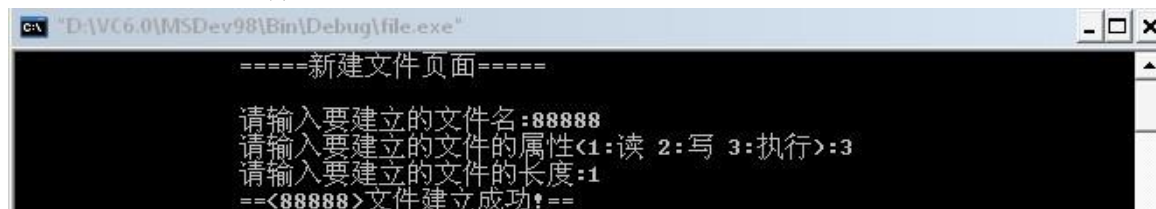
使用 open 命令打开一个文件，注意：对文件的所有读写等操作必须先打开文件，否则提示出错：



```
<<文件系统演示>>

*****
create 新建文件.
delete 删除文件.
open   打开文件.
close  关闭文件.
read   读出文件.
write  写进文件.
display 显示用户列表.
quit   安全退出.
*****
操作命令:create delete open close read write display quit
请输入操作命令:open
```

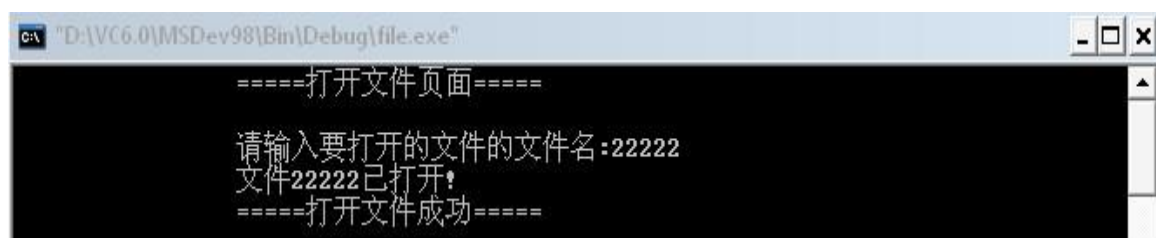
在这里测试新建文件 88888：



```
=====新建文件页面=====

请输入要建立的文件名:88888
请输入要建立的文件的属性<1:读 2:写 3:执行>:3
请输入要建立的文件的长度:1
==<88888>文件建立成功!==
```

在这里测试打开文件 22222：



```
=====打开文件页面=====

请输入要打开的文件的文件名:22222
文件22222已打开!
=====打开文件成功=====
```

当文件不存在用户目录下时，提示出错：



```
C:\ "D:\VC6.0\MSDev98\Bin\Debug\file.exe"

====打开文件页面====

请输入要打开的文件的文件名:56
打不到要文件打开的文件.打开文件失败!
```

打开文件后，使用 read 命令读入文件，如果文件还没有在上一步打开，则操作提示出错：

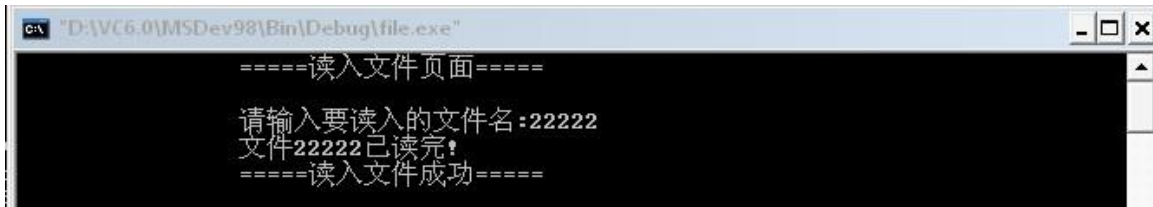


```
C:\ "D:\VC6.0\MSDev98\Bin\Debug\file.exe"

====读入文件页面====

请输入要读入的文件名:656
操作错误, 当前文件?还没有打开!
```

成功读入文件 22222:

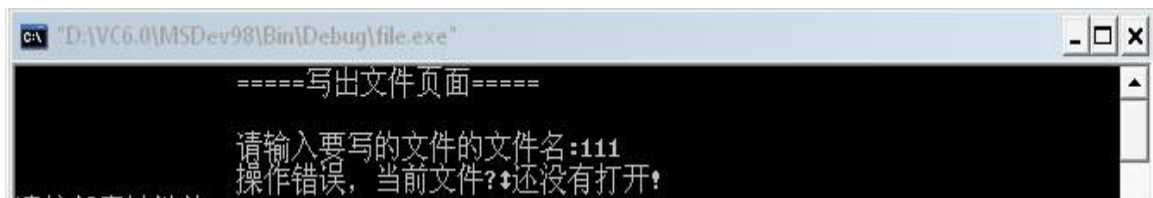


```
C:\ "D:\VC6.0\MSDev98\Bin\Debug\file.exe"

====读入文件页面====

请输入要读入的文件名:22222
文件22222已读完!
====读入文件成功====
```

同样，使用 write 写出文件，如果文件也没有被打开则提示出错：

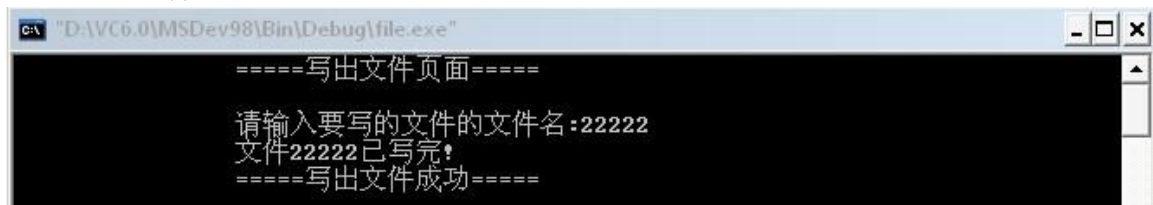


```
C:\ "D:\VC6.0\MSDev98\Bin\Debug\file.exe"

====写出文件页面====

请输入要写的文件的文件名:111
操作错误, 当前文件?还没有打开!
```

成功写出文件 22222



```
C:\ "D:\VC6.0\MSDev98\Bin\Debug\file.exe"

====写出文件页面====

请输入要写的文件的文件名:22222
文件22222已写完!
====写出文件成功====
```

使用 close 命令关闭文件，如果文件没有被打开则提示出错：




```
C:\ "D:\VC6.0\MSDev98\Bin\Debug\file.exe"

====关闭文件页面====

请输入要关闭的文件名:77777
文件77777没有打开!关闭失败!
```

成功关闭文件 22222:



```
C:\ "D:\VC6.0\MSDev98\Bin\Debug\file.exe"

====关闭文件页面====

请输入要关闭的文件名:22222
文件22222已成功关闭!
====关闭文件成功====
```

使用 delete 命令删除文件，找不到文件名字则提示出错：

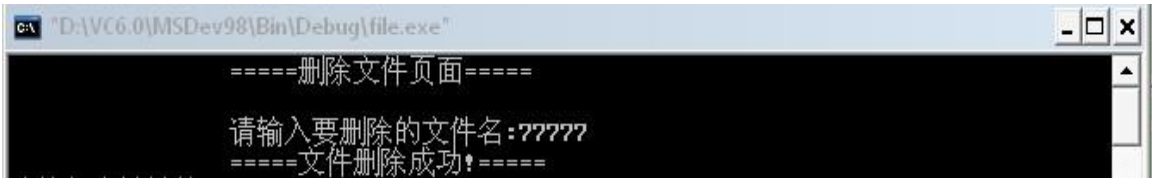


```
C:\ "D:\VC6.0\MSDev98\Bin\Debug\file.exe"

=====删除文件页面=====

请输入要删除的文件名:s
找不到要删除的文件!
```

成功删除用户一个文件 77777：

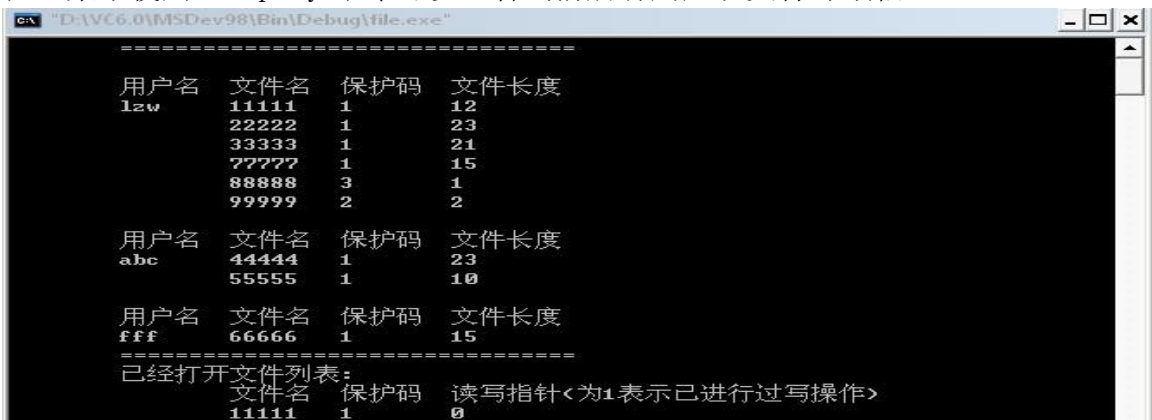


```
C:\ "D:\VC6.0\MSDev98\Bin\Debug\file.exe"

=====删除文件页面=====

请输入要删除的文件名:77777
=====文件删除成功!=====
```

在主菜单使用 display 命令可以查看当前所有用户和文件详细信息：



```
C:\ "D:\VC6.0\MSDev98\Bin\Debug\file.exe"

=====
用户名 文件名 保护码 文件长度
lzw 11111 1 12
    22222 1 23
    33333 1 21
    77777 1 15
    88888 3 1
    99999 2 2

用户名 文件名 保护码 文件长度
abc 44444 1 23
    55555 1 10

用户名 文件名 保护码 文件长度
fff 66666 1 15
=====
已经打开文件列表:
  文件名 保护码 读写指针<为1表示已进行过写操作>
  11111 1 0
=====
```

使用 quit 命令退出程序，则系统将自动保存当前操作结果：



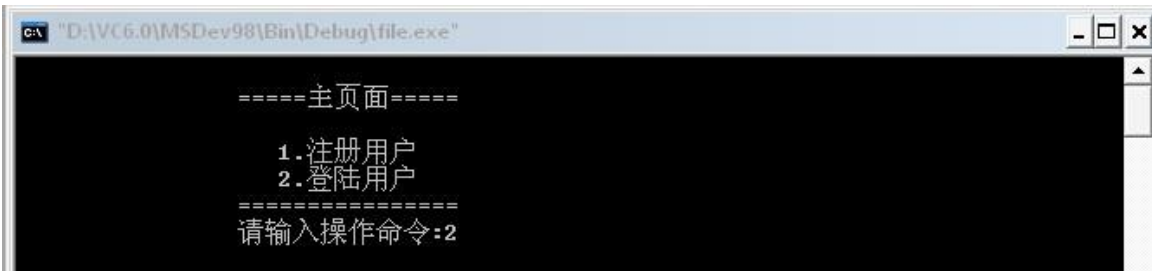
```
C:\ "D:\VC6.0\MSDev98\Bin\Debug\file.exe"

<<文件系统演示>>

*****
create 新建文件.
delete 删除文件.
open   打开文件.
close  关闭文件.
read   读出文件.
write  写进文件.
display 显示用户列表.
quit   安全退出.
*****

操作命令:create delete open close read write display quit
请输入操作命令:quit
当前目录信息已自动保存至文件database!
```

第二次运行程序，可以查看目录是否存在 database.txt 文件，如果存在直接使用登录功能可以获取上次的用户和文件信息：



```
C:\ "D:\VC6.0\MSDev98\Bin\Debug\file.exe"

=====主页面=====

1.注册用户
2.登陆用户
=====
请输入操作命令:2
```

登陆详细信息展示：


```

D:\VC6.0\MSDev98\Bin\Debug\file.exe

用户名 文件名 保护码 文件长度
lzw 11111 1 12
    22222 1 23
    33333 1 21
    77777 1 15
    88888 3 1
    99999 2 2

用户名 文件名 保护码 文件长度
abc 44444 1 23
    55555 1 10

用户名 文件名 保护码 文件长度
fff 66666 1 15

=====登陆页面=====

请输入登录用户名:lzw

=====用户lzw目录下的文件=====

11111
22222
33333
77777
88888
99999
请按任意键继续. . .
```

六、调试总结及心得体会

通过自己的理解，添加了注册和登陆的功能，只要一次保存了数据下一次就可以直接读取，容易测试，我是采用了字符串匹配的方式简介实现了文件操作的几个命令。文件保护码之类的设计这方面自己存有很大的疑惑尚未解决，所以在文件权限方面做得比较差，唯有通过进一步学习，解决了心中的疑惑后，再回过头来完善代码。

七、源程序代码：

```
#include<stdio.h>
#include<stdlib.h>
#include<iostream>
#define NULL 0
#define getpch(type,num) (type*)malloc(sizeof(type)*num)

struct ufd{
    /* 第二级：用户文件目录 UFD */
    char filename[10]; /* 文件名 */
    char procode[3]; /* 保护码 1:读 2:写 3:执行*/
    int length; /* 文件长度 */
}*p;

typedef struct ufd UFD;

struct mdf{
    /* 第一级：主目录 MDF */
    char name[10]; /* 用户名 */
    UFD* directory; /* 文件目录指针 */
    }maindir[10]; /* 用户数组 */
```



```

typedef struct mdf MDF;

struct afd{
    /* 文件运行目录 */
    char filename[10];    /* 文件名 */
    char procode[3];      /* 保护码 1:读 2:写 3:执行*/
    int rw;               /* 读写指针 初始状态读:0 写:文件长度 */
}afd[5];

int usernum, opennum;    /* 登陆用户数量 打开文件数量*/
int filenum[10];         /* 所有的用户目录下的文件数目*/
char username[10];       /* 当前登陆系统的用户名*/
char command[][10]={ "create", "delete", "open", "close", "read", "write", "display", "quit" };
/*文件操作命令*/

void login() /*用户注册，信息写入database文件*/
{
    int i, j;
    FILE *fp;
    int loginum;
    int creatnum[10];
    printf("\t\t====注册页面====\n\n");
    printf("\t\t请输入注册的用户数量:");
    scanf("%d", &loginum);
    for(i=0; i<loginum; i++)
    {
        printf("\t\t请输入第%d个用户名:", i+1);
        scanf("%s", maindir[i].name);
    }
    for(i=0; i<loginum; i++)
    {
        printf("\t\t请输入%s用户新建文件数(小于10):", maindir[i].name);
        scanf("%d", &creatnum[i]);
        p=getpch(UFD, 10);
        maindir[i].directory=p;
        for(j=0; j<creatnum[i]; j++)
        {
            printf("\t\t请输入新建文件名:");
            scanf("%s", p[j].filename);
            printf("\t\t请输入要建立的文件的属性(1:读, 2:写, 3:执行.):");
            scanf("%s", p[j].procode);
            printf("\t\t请输入要构造的文件长度:");

```

```

        scanf("%d",&p[j].length);
    }
}

fp=fopen("database.txt","wb");
fwrite(&loginum,sizeof(int),1,fp);          /* 写已注册用户数量到dat文件*/
for(i=0;i<loginum;i++)
{
    fwrite(maindir[i].name,sizeof(char)*10,1,fp); /* 写已注册用户名到dat文件*/
}
for(i=0;i<loginum;i++)
{
    fwrite(&creatnum[i],sizeof(int),1,fp);      /* 写用户新建文件数量到dat文件*/
    p=maindir[i].directory;                      /* UFD获得文件目录指针*/
    fwrite(p,sizeof(UFD),10,fp);                /* 写用户文件目录指针地址到dat文
件*/
}
fclose(fp);
printf("\n\t\t====注册完毕====\n");
printf("\n\t\t");
system("pause");
system("cls");
}

void init()    /*初始化读入login信息*/
{
    int i,j;
    FILE *fp;
    fp=fopen("database.txt","rb");
    if(fp==NULL)
    {
        printf("\n\t警告:根目录不存在记录,请先注册用户\n");
        login();
        fp=fopen("database.txt","rb");
    }
    fread(&usernum,sizeof(int),1,fp);          /* 读入已注册用户数量*/
    for(i=0;i<usernum;i++)
    {
        fread(maindir[i].name,sizeof(char)*10,1,fp); /* 读入已注册用户名*/
        p=getpch(UFD,10);
        maindir[i].directory=p;
    }
    for(i=0;i<usernum;i++)
    {

```

```

        fread(&j, sizeof(int), 1, fp);
        filenum[i]=j;                                /*读入用户新建文件数量*/
        p=maindir[i].directory;                      /*获得文件目录指针*/
        fread(p, sizeof(UFD), 10, fp);              /*读入用户文件目录指针地址p*/
    }
    opennum=0;                                        /*运行文件数*/
}

void createfile(int userid)                          /*新建文件*/
{
    system("cls");
    char cfilename[10];
    int filelen, i;
    char code[3];
    if(filenum[userid]>=10) /*如果用户的文件已有10个*/
    {
        printf("\n\t\t用户的文件数量已达到上限, 即10个!\n");
        return ;
    }
    printf("\t\t====新建文件页面====\n\n");
    printf("\t\t请输入要建立的文件名:");
    scanf("%s", cfilename);
    for(i=0; i<filenum[userid]; i++) /*检查是否有同名文件, 有则提示然后返回*/
    {
        if(strcmp((p+i)->filename, cfilename)==0)
        {
            printf("\t\t文件名%s已存在!\n", cfilename);
            system("pause");
            system("cls");
            return ;
        }
    }
    printf("\t\t请输入要建立的文件的属性(1:读 2:写 3:执行):");
    scanf("%s", code);
    printf("\t\t请输入要建立的文件长度:");
    scanf("%d", &filelen);
    strcpy(p[filenum[userid]].filename, cfilename); /*添加新文件名*/
    strcpy(p[filenum[userid]].procode, code);      /*添加保护码*/
    p[filenum[userid]].length=filelen;              /*添加文件长度*/
    filenum[userid]=filenum[userid]+1;             /*更新文件数量*/
    printf("\t\t==<%s>文件建立成功!==\n", cfilename);
    system("pause");
    system("cls");
}

```

```

void deletefile(int userid)          /*删除文件*/
{
    system("cls");
    char dfilename[10];
    int i, j;
    printf("\t\t====删除文件页面====\n\n");
    printf("\t\t请输入要删除的文件名:");
    scanf("%s", dfilename);
    for(i=0; i<filenum[userid]; i++)
    {
        if(strcmp(p[i].filename, dfilename)==0) /*找到要删除的文件*/
        {
            for(j=i; j<filenum[userid]-1; j++)
            {
                strcpy(p[j].filename, p[j+1].filename);
                strcpy(p[j].procode, p[j+1].procode);
                p[j].length=p[j+1].length;
                if(j==filenum[userid]-1)
                {
                    strcpy(p[j+1].filename, " ");
                    strcpy(p[j+1].procode, " ");
                    p[j+1].length=0;
                    free(p+j+1);
                }
            }

            filenum[userid]--;
            printf("\t\t====文件删除成功!====\n");
            system("pause");
            system("cls");
            return ;
        }
    }

    printf("\t\t找不到要删除的文件!\n");
    system("pause");
    system("cls");
}

void openfile(int userid)          /*打开文件*/
{
    system("cls");
    int i;
    char ofilename[10];

```

```

printf("\t\t====打开文件页面====\n\n");
if(opennum>=5)
{
    printf("\t\t打开超过5个文件!\n");
    return ;
}
printf("\t\t请输入要打开的文件的文件名:");
scanf("%s", ofilename);
for(i=0; i<opennum; i++)
{
    if(strcmp(afd[i]. filename, ofilename)==0) /*文件已经打开*/
    {
        printf("\t\t文件%s已打开!\n", ofilename);
        return ;
    }
}
for(i=0; i<filenum[userid]; i++) /*在运行文件目录中找到要打开的文件*/
{
    if(strcmp(p[i]. filename, ofilename)==0) /*找到该文件*/
    {
        strcpy(afd[opennum]. filename, ofilename); /*文件运行目录添加打开文件名*/
        strcpy(afd[opennum]. procode, p[i]. procode); /*文件运行目录添加文件保护码*/
        afd[opennum]. rw=0; /*更新文件运行目录文件读写指针*/
        opennum++;
        printf("\t\t文件%s已打开!\n", ofilename);
        printf("\t\t====打开文件成功====\n\n");
        system("pause");
        system("cls");
        return ;
    }
}
printf("打不到要文件打开的文件, 打开文件失败!\n");
system("pause");
system("cls");
}

void closefile() /*关闭文件*/
{
    system("cls");
    int i, j;
    char clfilename[10];
    if(opennum==0) /*没有文件被打开过*/
    {
        printf("\t\t当前还没有打开文件!\n");
    }
}

```

```

        return ;
    }
    printf("\t\t====删除文件页面====\n\n");
    printf("\t\t请输入要关闭的文件名:");
    scanf("%s", clfilename);
    for (i=0; i<opennum; i++)
    {
        if (strcmp(afd[i].filename, clfilename)==0) /*在运行文件目录中找到要关闭的文件*/
        {
            if (afd[i].rw==1)
            {
                printf("\t\t文件写回磁盘!\n");
            }
            for (j=i; j<opennum-1; j++) /*更新文件运行目录, 关闭文件*/
            {
                strcpy(afd[j].filename, afd[j+1].filename);
                strcpy(afd[j].procode, afd[j+1].procode);
                afd[j].rw=afd[j+1].rw;
            }
            opennum--;
            printf("\t\t文件%s已成功关闭!\n", clfilename);
            printf("\t\t====关闭文件成功====\n\n");
            system("pause");
            system("cls");
            return ;
        }
    }
    printf("\t\t文件%s没有打开!关闭失败!\n", clfilename);
    system("pause");
    system("cls");
}

void readfile() /*读入文件*/
{
    system("cls");
    int i;
    char rfilename[10];
    if (opennum==0) /*没有文件被打开过*/
    {
        printf("\t\t操作错误, 当前还没有打开文件!\n");
        system("pause");
        system("cls");
        return ;
    }
}

```

```

printf("\t\t====读入文件页面====\n\n");
printf("请输入要读入的文件名:");
scanf("%s", rfilename);
for(i=0; i<opennum; i++)
{
    if(strcmp(afd[i].filename, rfilename)==0) /*在运行文件目录中找到要读的文件*/
    {
        if(afd[i].procode[0]!='1') /*文件没有读权限, 将不执行读操作, 提示返回*/
        {
            printf("\t\t权限不足, 该文件不能读! 读失败!\n");
            return ;
        }
        printf("\t\t文件%s已读完!\n", rfilename);
        printf("\t\t====读入文件成功====\n\n");
        system("pause");
        system("cls");
        break ;
    }
}

}

void writefile() /*写出文件*/
{
    system("cls");
    int i;
    char wfilename[10];
    if(opennum==0) /*没有文件被打开过*/
    {
        printf("\t\t当前还没有打开文件!\n");
        system("pause");
        system("cls");
        return ;
    }
    printf("\t\t====写出文件页面====\n\n");
    printf("\t\t请输入要写的文件的文件名:");
    scanf("%s", wfilename);
    for(i=0; i<opennum; i++)
    {
        if(strcmp(afd[i].filename, wfilename)==0) /*在运行文件目录中找到要关闭的文件*/
        {
            if(p[i].procode[1]!='1') /*文件没有写权限, 将不执行写操作, 提示返回*/
            {
                printf("\t\t该文件不能写! 写失败!\n");
                return ;
            }

```



```

    }
    afd[i].rw=1;
    printf("\t\t文件%s已写完!\n",wfilename);
    printf("\t\t====写出文件成功====\n\n");
    return ;
}
}
printf("\t\t操作错误，当前文件%s还没有打开!\n");
system("pause");
system("cls");
}

void displayinfo()    /*显示所有用户目录文件信息*/
{
    system("cls");
    int i, j;
    printf("\t===== \n");
    for(i=0; i<usernum; i++)    /*登陆页面用户列表显示*/
    {
        printf("\n\t用户名\t文件名\t保护码\t文件长度\n");
        printf("\t%s", maindir[i].name);
        for(j=0; j<filenum[i]; j++)
        {
            if(j!=0)printf("\t");

printf("\t%s\t%s\t%d\n", maindir[i].directory[j].filename, maindir[i].directory[j].procode, maindir[i].directory[j].length);
        }
    }
    printf("\t===== \n");
    printf("\t已经打开文件列表:\n\t\t\t文件名\t保护码\t读写指针(为1表示已进行过写操作)\n");
    for(i=0; i<opennum; i++)
    {
        printf("%s\t%s\t%d\n", afd[i].filename, afd[i].procode, afd[i].rw);
    }
    system("pause");
    system("cls");
}

void savefile()    /*保存操作*/
{
    /*开始把输入的构造信息写入到文件dat*/
    int i;

```

```

FILE *fp;
fp=fopen("database.txt","wb");
fwrite(&usernum,sizeof(int),1,fp);
for(i=0;i<usernum;i++)
{
    fwrite(maindir[i].name,sizeof(char)*10,1,fp);
}
for(i=0;i<usernum;i++)
{
    fwrite(&filenum[i],sizeof(int),1,fp);
    p=maindir[i].directory;
    fwrite(p,sizeof(UFD),10,fp);
}
fclose(fp);
printf("\t\t当前目录信息已自动保存至文件database!\n");
}

```

```

void main()
{
    int i,j,n,userid,choose=0,quit=0;
    char inputname[10];
    system("cls");
    printf("\n\t  ----10级计算机科学与技术03班----");
    printf("\n\t  -----计科3张大川-----\n");
    printf("\n\t\t====主页面====\n");
    printf("\n\t\t  1. 注册用户");
    printf("\n\t\t  2. 登陆用户");
    printf("\n\t\t===== \n");
    printf("\t\t请输入操作命令:");
    scanf("%d",&n);
    if(n==1)
    {
        login();    /*用户注册，用户文件信息写入dat文件*/
    }
    init();    /*信息初始化*/
    system("cls");
    for(i=0;i<usernum;i++)    /*登陆页面用户列表显示*/
    {
        printf("\n\t用户名\t文件名\t保护码\t文件长度\n");
        printf("\t\t%s",maindir[i].name);
        for(j=0;j<filenum[i];j++)
        {
            if(j!=0)printf("\t");

```

```

printf("\t%s\t%s\t%d\n", maindir[i].directory[j].filename, maindir[i].directory[j].procode, maindir[i].directory[j].length);
    }
}

printf("\n\t\t====登陆页面====\n\n");
j=0;
while(j==0)
{
    printf("\n\t\t请输入登录用户名:");          /*用户登陆*/
    scanf("%s", username);
    for(userid=0;userid<usenum;userid++)
    {
        if(strcmp(maindir[userid].name, username)==0) /*用户存在*/
        {
            j=1;
            break;
        }
    }
    if(j==1)
    {
        break;
    }
    printf("\n\t\t输入的用户名不存在!\n");          /*用户不存在*/
}

p=maindir[userid].directory;          /*获得用户文件目录指针*/
printf("\n\t\t====用户%s目录下的文件====\n", username);
for(i=0;i<filenum[userid];i++)
{
    printf("\n\t\t%s", p[i].filename);          /*显示该用户文件列表*/
}

printf("\n\t\t");
system("pause");
system("cls");
opennum=0;
while(true)          /*操作主目录*/
{
    j=0;
    system("cls");
    printf("\n\t\t<<文件系统演示>>\n");
    printf("\n\t\t*****\t\t");
    printf("\n\t\t\t create   新建文件.");
    printf("\n\t\t\t delete  删除文件.");
    printf("\n\t\t\t open    打开文件.");
}

```

```
printf("\n\t\t\t close   关闭文件.");
printf("\n\t\t\t read    读出文件.");
printf("\n\t\t\t write   写进文件.");
    printf("\n\t\t\t display 显示用户列表.");
printf("\n\t\t\t quit    安全退出.");
printf("\n\t\t*****\t\t\n");
    printf("\t\t操作命令:create delete open close read write display quit\n");
printf("\t\t请输入操作命令:");
    scanf("%s", inputname);
    getchar();
for(i=0;i<8;++i)
{
    if(strcmp(command[i], inputname)==0) /*命令存在*/
    {
        j=1;
        choose=i+1;
        break;
    }
}
if(j==0)
{
    printf("输入命令不正确!");
    system("pause");
    continue;
}

switch(choose)
{

case 1:
    createfile(userid);
    break;

case 2:
    deletetfile(userid);
    break;

case 3:
    openfile(userid);
    break;

case 4:
    closefile();
```

```
        break;

    case 5:
        readfile();
        break;

    case 6:
        writefile();
        break;

    case 7:
        displayinfo();
        break;

    case 8:
        quit=1;
        break;

    default:
        break;
}

    if(quit==1)break;
}

savefile();
}
```