实验六：文件系统

一、目的要求

1、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。

2、要求设计一个 n个用户的文件系统，每次用户可保存m个文件，用户在一次运行中只能打开一个文件，对文件必须设置保护措施，且至少有Create、delete、open、close、read、write等命令。

二、例题：

　①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。

②程序采用二级文件目录（即设置主目录[MFD]）和用户文件目录（UED）。另外，为打开文件设置了运行文件目录（AFD）。

③为了便于实现，对文件的读写作了简化，在执行读写命令时，只需改读写指针，并不进行实际的读写操作。

④ 算法与框图：

a、因系统小，文件目录的检索使用了简单的线性搜索。

b、文件保护简单使用了三位保护码：允许读写执行、对应位为 1，对应位为0，则表示不允许读写、执行。

c、程序中使用的主要设计结构如下：

Ⅰ、主文件目录和用户文件目录（ MFD、UFD）

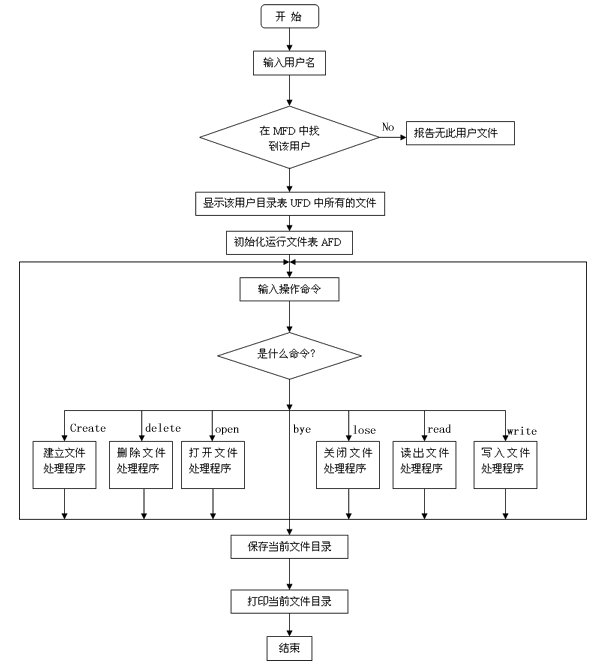
Ⅱ、打开文件目录（ AFD）（即运行文件目录）

|  |
| --- |
| M D F |
| 用户名 |
| 文件目录指针 |
| 用户名 |
| 文件目录指针 |
|  |

|  |
| --- |
| U F D |
| 文件名 |
| 保护码 |
| 文件长度 |
| 文件名 |
|  |

|  |
| --- |
| A F D |
| 打开文件名 |
| 打开保护码 |
| 读写指针 |

三、调度算法的流程图



四、文件管理源程序

#include<stdio.h>

#include<string.h>

#include<malloc.h>

#include<stdlib.h>

typedef struct ufd

{ char filename[10];/\*文件名\*/

char procode[8];/\*属性\*/

int length;/\*文件长度\*/

struct ufd \*nextfile;/\*指向下一个文件\*/

}UFD;

typedef struct mfd

{ char username[10];/\*用户名\*/

struct ufd \*link;/\*指向该用户的第一个文件\*/

}MFD;

typedef struct protected\_flag

{ char code[4];

}PRO;

typedef struct afd/\*运行文件目录\*/

{ char filename[10];/\*打开文件名\*/

char procode[4];

int rwpointer;/\*读写指针\*/

}AFD;

PRO flag[3]={"100",/\*只读\*/

"110",/\*读写\*/

"001"/\*可执行\*/

};

UFD \*rw\_pointer;/\*读写指针\*/

AFD \*afd=NULL;

MFD filesystem[10];

int num;/\*当前用户个数\*/

void displayallfile()

{ int i;

UFD \*p;

for(i=0;i<num;i++)

{

printf("用户 :%s 目录如下!\n",filesystem[i].username);

p=filesystem[i].link;

while(p)

{ printf("\t文件名: %s\t||",p->filename);

printf("文件属性: %s\t||",p->procode);

printf("文件长度: %d\n\n",p->length);

p=p->nextfile;

}

}

}

void input() /\*建立用户文件系统\*/

{ int i,j;

int length;

char filename[10];

char procode[4];

UFD \*p1,\*p2,\*first;

for(i=0;i<10;i++)

filesystem[i].link=NULL;

printf("用户号码:\n");

scanf("%d",&num);

for(i=0;i<num;i++)/\*输入各用户名称\*/

{

printf("输入第 %d个用户名:\n",i+1);

scanf("%s",filesystem[i].username);}

for(i=0;i<num;i++)/\*输入某用户的文件数据\*/

{

j=1;

printf(" 第%d个用户的文件如下\n",i+1);

printf("文件 %d:\n",j);

printf("长度 :");

scanf("%d",&length);

if(!length)

continue;

printf("文件名:");

scanf("%s",filename);

printf("文件属性(100只读，110读写，001可执行)");

scanf("%s",procode);

first=(UFD\*)malloc(sizeof(UFD));

first->nextfile=NULL;

strcpy(first->filename,filename);

strcpy(first->procode,procode);

first->length=length;

p1=p2=first;

j++;

while(1)

{printf("文件 %d:\n",j);

printf("长度 :");

scanf("%d",&length);

if(!length)

break;

printf("文件名:");

scanf("%s",filename);

printf("属性(100只读，110读写，001可执行):");

scanf("%s",procode);

p2=(UFD\*)malloc(sizeof(UFD));

p2->nextfile=NULL;

strcpy(p2->filename,filename);

strcpy(p2->procode,procode);

p2->length=length;

j++;

p1->nextfile=p2;

p1=p1->nextfile; }

filesystem[i].link=first; }

displayallfile();

}

int findusername(char \* username)/\*返回用户名下标,当I=NUM时，用户不存在\*/

{ int i;

i=0;

while(i<num)

{if(!strcmp(filesystem[i].username ,username))

break;

i++;}

return i;

}

UFD \* findfilename(char \*filename,UFD \*filepointer)

{UFD \*k;

k=filepointer;

while(k&&strcmp(filename,k->filename))

k=k->nextfile;

return k;}

void displayfile(UFD \*link)

{ UFD \*p;

p=link;

if(!p)

{ printf("\n此目录没有该文件！");

return;}

printf("\n用户文件如下:\n");

while(p)

{ printf("%s\t||",p->filename);

printf("%s\t||",p->procode);

printf("%d\n\n",p->length);

p=p->nextfile;}

}

void createfile()

{ char username[10];

char filename[10];

int i;

UFD \*p1,\*p2,\*newfile;

if(num>=10)/\*用户文件目录已满，创建文件失败\*/

{

printf("用户文件目录已满\n创建文件失败!\n");

return;}

printf("输入用户名:\n");

scanf("%s",username);

i=findusername(username);/\*返回用户名下标\*/

if(i==num)/\*如果该用户不存在，且用户个数不小于10，先创建用户再建立用户文件\*/

{printf("用户不存在，请建立用户!\n");

strcpy(filesystem[num].username ,username);/\*创建用户目录\*/

num++;

{newfile =(UFD \*)malloc(sizeof(UFD));

newfile->nextfile=NULL;

filesystem[num-1].link=newfile;

printf("文件名:");

scanf("%s",filesystem[num-1].link->filename);

printf("\n属性:");

scanf("%s",filesystem[num-1].link->procode);

printf("\n长度:");

scanf("%d",&(filesystem[num-1].link->length));

printf("\n文件已创建!\n");}

}

else/\*为已存在的用户建立文件\*/

{p1=p2=filesystem[i].link ;

printf("输入文件名:");

scanf("%s",filename);

while(p2)

{ if (!strcmp(p2->filename,filename))

{ printf("文件名已存在，请输入新文件名!\n");

scanf("%s",filename); }

p1=p2;

p2=p2->nextfile;

}

newfile=(UFD \*)malloc(sizeof(UFD));

newfile->nextfile=NULL;

strcpy(newfile->filename,filename);

printf("\n属性:");

scanf("%s",newfile->procode);

printf("\n长度:");

scanf("%d",&(newfile->length));

p1->nextfile=newfile;//

printf("\n文件已创建!\n");}

displayallfile();

}

void displayuser()/\*显示用户目录\*/

{int i;

if(num<=0)

return;

printf("\n用户目录如下! ");

for(i=0;i<num;i++);

printf("\n%s",filesystem[i].username);}

void deletefile() /\*删除文件\*/

{ int i;

char username[10];

char filename[10];

UFD \*p1,\*p2;

printf("\n输入用户名:\n");

scanf("%s",username);

i=findusername(username);//返回用户名下标

if(i>=num)

{printf("用户不存在!\n删除失败!");

return ;}

else

{ printf("\n输入文件名:");

scanf("%s",filename);

p1=p2=filesystem[i].link;

while(p2&&strcmp(p2->filename,filename))

{p1=p2;

p2=p2->nextfile;}

if(!p2)

printf("\n文件不存在!\n删除失败!\n");

else

{ if(p1==p2)

filesystem[i].link=p1->nextfile;

else

p1->nextfile=p2->nextfile;

delete(p2);

printf("文件已删除!\n");}

}

displayallfile();}

AFD \* openfile()

{ char username[10];

char filename[10];

int i;

UFD \*p=NULL;

AFD \*k=NULL;/\*为打开的文件设置了运行文件目录\*/

printf("输入用户名:\n");

scanf("%s",username);

i=findusername(username);/\*返回用户名下标\*/

if(i>=num)

{printf("\n用户不存在!\n不能打开文件!\n");

return NULL;}

else/\*用户存在\*/

{printf("\n输入文件名:");

scanf("%s",filename);

p=findfilename(filename,filesystem[i].link);

if(!p)/\*用户文件不存在\*/

{printf("\n文件不存在!\n");

return NULL;}

else{afd=(AFD \*)malloc(sizeof(AFD));

strcpy(afd->filename,p->filename);

strcpy(afd->procode,p->procode);

afd->rwpointer=0;

printf("\n文件已打开!\n");

rw\_pointer=p;/\*为了读文件保留指针\*/

printf("\t文件目录AFD:\n"); /\*打开运行文件目录AFD\*/

printf("\t文件名 : %s\n",afd->filename);

printf("\t属性 : %s\n",afd->procode);

printf("\t读写指针 : %d\n",afd->rwpointer); }

}

return afd; }

void closefile()

{ if(!afd)

{ printf("\n没有文件打开!\n");

return; }

delete(afd);

afd=NULL;

printf("\n文件已关闭!\n");}

void readfile()//

{ AFD \*p;

p=openfile();

if(!p)

return;

if(strcmp(p->procode,"100")&&strcmp(p->procode,"110"))

printf("\nfile:%s 文件不可读!\n",p->filename);/\*操作\*/

else

{ printf("已读取文件!\n");

printf("\t文件名:%s\n",p->filename);

printf("\t属性:%s\n",p->procode);

printf("\t读写指针:%d\n",rand()%rw\_pointer->length); }

closefile();/\*关闭文件\*/

}

void writefile()

{ AFD \*p;/\*运行文件目录\*/

UFD \*k;

p=openfile();

k=rw\_pointer;

if(!p)

return;

if(strcmp(p->procode,"110"))

{printf("\nfile:%s 该文件不能写!\n",p->filename);

return; }

else

{ p->rwpointer=rand()%k->length;

k->length=p->rwpointer ;/\*修改读书指针写回文件\*/

}

printf("\n已完成写操作!\n");

printf("\t文件名 : %s\n",k->filename);

printf("\t属性 : %s\n",k->procode);

printf("\t长度 : %d\n",k->length);

closefile();

}

void executefile()

{ AFD \*p;

p=openfile();

if(!p)

{printf("\n文件不存在!\n");

return;}

if(strcmp(p->procode,"001"))

{printf("\n该文件不能执行!\n");

printf("\n文件属性为不可执行文件!\n");

return;}

else{

closefile();

printf("\n文件执行完毕!"); }

}

void menu()

{ printf("------------------------------------------------------------------------------\n");

printf(" <文件管理> \n");

printf("------------------------------------------------------------------------------\n");

printf(" MENU :\n");

printf("\t\tmenu----------显示菜单\n");

printf("\t\topen----------打开文件\n");

printf("\t\tclose---------关闭文件\n");

printf("\t\tcreate--------创建文件\n");

printf("\t\tread----------读文件\n");

printf("\t\twrite---------写文件\n");

printf("\t\texecute-------执行文件\n");

printf("\t\tdelete--------删除文件\n");

printf("\t\tdisplay--------显示文件列表\n");

printf("\t\texit--------退出\n");

printf("\t注意：输入时当文件长度为0时表示该用户的文件输入结束!\n");

printf("------------------------------------------------------------------------------\n");

}

void fileoperation()/\*文件基本操作\*/

{ char str[10];

int sort;

while(1)

{printf("输入命令:\n");

printf(">>");

scanf("%s",str);

if(!strcmp(str,"create"))sort=1;

else

if(!strcmp(str,"delete"))sort=2;

else

if(!strcmp(str,"read"))sort=3;

else

if(!strcmp(str,"write"))sort=4;

else

if(!strcmp(str,"execute"))sort=5;

else

if(!strcmp(str,"open"))sort=6;

else

if(!strcmp(str,"close"))sort=7;

else

if(!strcmp(str,"menu"))sort=8;

else

if(!strcmp(str,"display"))sort=9;

else

if(!strcmp(str,"exit"))sort=10;

else

sort=0;

switch(sort)

{ case 1:createfile();break;

case 2:deletefile();break;

case 3:readfile();break;

case 4:writefile();break;

case 5:executefile();break;

case 6:openfile();break;

case 7:closefile();break;

case 8:menu();break;

case 9: displayallfile();break;

case 10: return;

default:break; }

}

}

void main()

{ menu();

input();

fileoperation();

}