# 8文件实验

**8.1实验目的**

1．熟悉文本文件和二进制文件在磁盘中的存储方式；

2．熟练掌握流式文件的读写方法。

**8.2实验题目及要求**

**8.2.1．文件类型的程序验证题**

设有程序：

#include <stdio.h>

int main(void)

{

short a=0x253f,b=0x7b7d;

char ch;

FILE \*fp1,\*fp2;

fp1=fopen("d:\\abc1.bin","wb+");

fp2=fopen("d:\\abc2.txt","w+");

fwrite(&a,sizeof(short),1,fp1);

fwrite(&b,sizeof(short),1,fp1);

fprintf(fp2,"%hx %hx",a,b);

rewind(fp1); rewind(fp2);

while((ch = fgetc(fp1)) != EOF)

putchar(ch);

putchar('\n');

while((ch = fgetc(fp2)) != EOF)

putchar(ch);

putchar('\n');

fclose(fp1);

fclose(fp2);

return 0;

}

（1）请思考程序的输出结果，然后通过上机运行来加以验证。

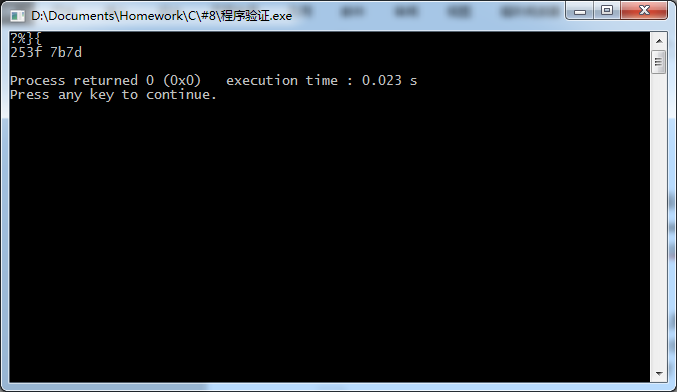
（2）将两处sizeof(short)均改为sizeof(char)结果有什么不同，为什么？

（3）将fprintf(fp2,"%hx %hx",a,b) 改为 fprintf(fp2,"%d %d",a,b)结果有什么不同。

（1）

**【分析】**fp1所指向的文件是以二进制形式输入的，而fp2指向的文件是以文本形式输入的，且把数字转换为了字符串，所以读出来的时候fp1要转换为相应的ASCII码，而fp2为照原样读出的，fp1输出为?%}{ ，fp2输出为253f 7b7d。

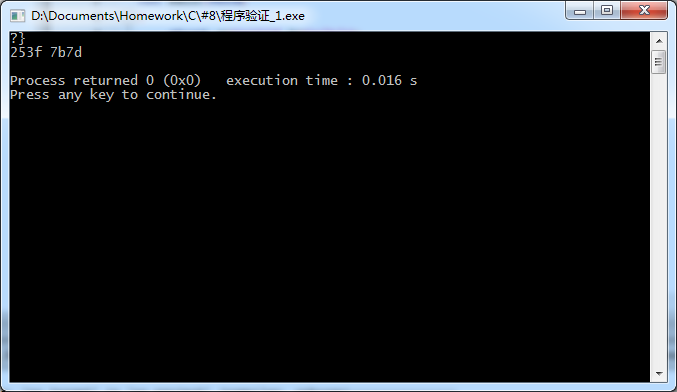
**【验证】**程序运行结果与设想一致



（2）

**【分析】**修改后只会向文件中写入前一个字节，输出也只会输出前一个字节对应的ASCII值

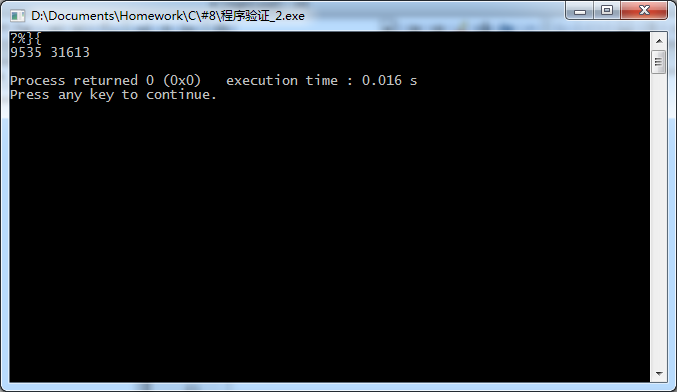
**【验证】**输出与分析结果一致



（3）

**【分析】**写入文本时转换为了十进制，输出时也会输出相应的十进制数

**【验证】**输出与分析结果一致



**8.2.2．源程序修改替换题**

将指定的文本文件内容在屏幕上显示出来，命令行的格式为：

type filename

(1) 源程序中存在什么样的逻辑错误（先观察执行结果）？对程序进行修改、调试，使之能够正确完成指定任务。

#include<stdio.h>

#include<stdlib.h>

int main(int argc, char\* argv[])

{

char ch;

FILE \*fp;

if(argc!=2){

printf("Arguments error!\n");

exit(-1);

}

if((fp=fopen(argv[1],"r"))==NULL){ /\* fp 指向 filename \*/

printf("Can't open %s file!\n",argv[1]);

exit(-1);

}

while(ch=fgetc(fp)!=EOF) /\* 从filename中读字符 \*/

putchar(ch); /\* 向显示器中写字符 \*/

fclose(fp); /\* 关闭filename \*/

return 0;

}

（2）用输入输出重定向freopen改写main函数。

（1）

**【分析】**源程序while(ch=fgetc(fp)!=EOF)中!=的优先级高于=，因此会出错。

**【替换】**

#include<stdio.h>

#include<stdlib.h>

int main(int argc, char\* argv[])

{

char ch;

FILE \*fp;

if(argc!=2){

printf("Arguments error!\n");

exit(-1);

}

if((fp=fopen(argv[1],"r"))==NULL){ /\* fp 指向 filename \*/

printf("Can't open %s file!\n",argv[1]);

exit(-1);

}

while((ch=fgetc(fp))!=EOF) /\* 从filename中读字符 \*/

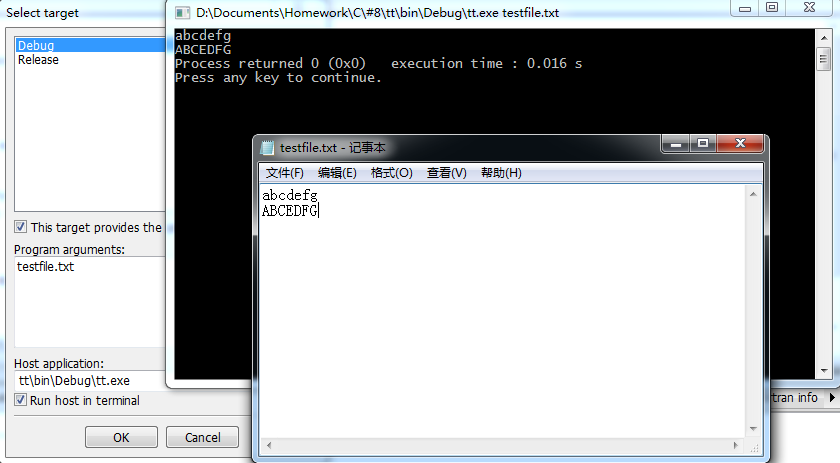
putchar(ch); /\* 向显示器中写字符 \*/

fclose(fp); /\* 关闭filename \*/

return 0;

}

**【测试】**修改后能正确输出文件中的内容



（2）

**【分析】**将stdin重定向到testfile.txt即可

**【替换】**

#include<stdio.h>

#include<stdlib.h>

int main(int argc, char\* argv[])

{

char ch;

FILE \*fp;

if(argc!=2){

printf("Arguments error!\n");

exit(-1);

}

if((fp=fopen(argv[1],"r"))==NULL){ /\* fp 指向 filename \*/

printf("Can't open %s file!\n",argv[1]);

exit(-1);

}

freopen(argv[1],"r",stdin);

while((ch=getchar())!=EOF) /\* 从filename中读字符 \*/

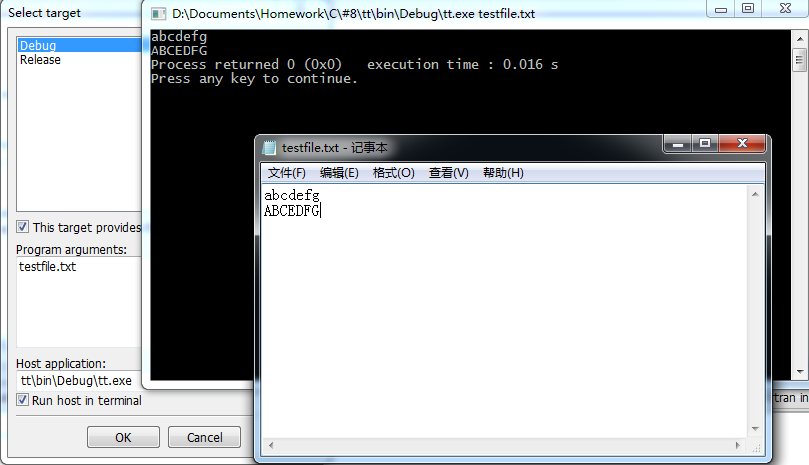
putchar(ch); /\* 向显示器中写字符 \*/

fclose(fp); /\* 关闭filename \*/

return 0;

}

**【测试】**流重定向成功



**8.2.3．编程设计题**

（1）从键盘输入一行英文句子，将每个单词的首字母换成大写字母，然后输出到一个磁盘文件“test”中保存。

(2) 设计一个通信录管理程序，程序应有以下9个选项：

① 录入每个联系人的基本信息（至少应有姓名、单位、电话、邮件地址）；

② 从磁盘文件中读取记录到内存；

③ 保存记录到磁盘文件；

④ 修改记录；

⑤ 插入一条记录；

⑥ 删除一条记录；

⑦ 显示所有的记录；

⑧ 按姓名对记录升序排序

⑨ 退出

要求：

1) 用链表结构；

1) 采用结构化程序设计，将每个功能定义成函数；

2) 应有用户界面，提供菜单选项。

（1）

**【分析】**字符串处理加上文件操作（文本写入）

**【程序】**

#include<stdio.h>

#include<string.h>

int main(void){

char sentence[65535];

FILE \*outfile;

outfile = fopen("test","w");

fgets(sentence,65534,stdin);

for(int i=0;sentence[i]!='\0';i++){

if(i==0 && 'a'<=sentence[i] && sentence[i]<='z')

sentence[i]+=('A'-'a');

if(sentence[i]==' ' && 'a'<=sentence[i+1] && sentence[i+1]<='z')

sentence[i+1]+=('A'-'a');

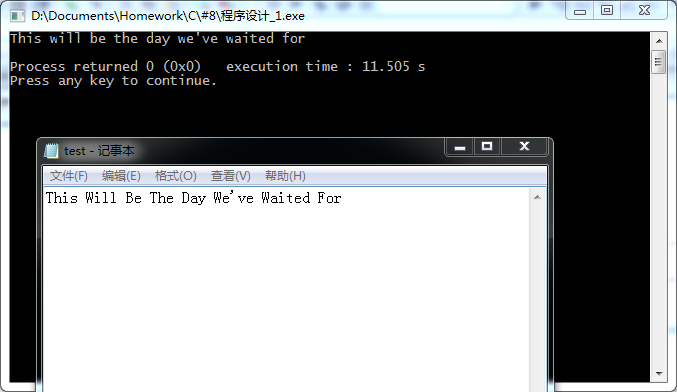
}

fwrite(sentence,sizeof(char),strlen(sentence),outfile);

return 0;

}

**【测试】**输入This will be the day we’ve waited for，观察文件中每个单词的首字母有无被转换为大写。



结果与预想一致。

（2）

**【分析】**使用链表加上文件操作，由于只要求按照姓名排序，座椅在插入的同时进行排序即可，省去排序的程序。

**【程序】**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct node{

char name[81];

char comp[101];

char tel[21];

char mail[101];

struct node \*next;

}Node;

void read(Node \*head);

void insert(Node \*head);

void modify(Node \*head);

void delet(Node \*head);

void showall(Node \*head);

void exiting(Node \*head);

int main(void){

Node \*head = (Node \*)malloc(sizeof(Node));

(head->name)[0]=0;

head->next=NULL;

int choose;

for(;;){

printf("\nchoose from the options:\n"

"1.read from file\n"

"2.modify\n"

"3.insert\n"

"4.delete\n"

"5.show all\n"

"6.sort by name\n"

"7.save and exit\n\n");

scanf("%d",&choose);

getchar();

switch(choose){

case 1:

read(head);

break;

case 2:

modify(head);

break;

case 3:

insert(head);

break;

case 4:

delet(head);

break;

case 5:

showall(head);

break;

case 6:

showall(head);

break;

case 7:

exiting(head);

break;

}

}

}

void read(Node \*head){

FILE \*data = fopen("data.dat","rb");

Node \*temp = head;

for(;fread(head,sizeof(Node),1,data)==1;head = head->next)

head->next = (Node \*)malloc(sizeof(Node));

for(;temp->next!=head;temp = temp->next);

temp->next=NULL;

free(head);

fclose(data);

}

void exiting(Node \*head){

FILE \*data = fopen("data.dat","wb");

for(;head!=NULL;head=head->next)

fwrite(head,sizeof(Node),1,data);

fputc(EOF,data);

fclose(data);

exit(0);

}

void insert(Node \*head){

Node ins;

printf("input name, mail, tel, comp:\n");

scanf("%s %s %s %s",ins.name,ins.mail,ins.tel,ins.comp);

Node \*temp = head;

for(;head!=NULL && (head->name)[0]<=(ins.name)[0];head = head->next);

for(;temp->next!=head;temp = temp->next);

temp->next = (Node \*)malloc(sizeof(Node));

temp = temp->next;

temp->next = head;

strcpy(temp->name,ins.name);

strcpy(temp->mail,ins.mail);

strcpy(temp->tel,ins.tel);

strcpy(temp->comp,ins.comp);

}

void modify(Node \*head){

Node modi;

char name[81];

printf("input name to modify:\n");

scanf("%s",name);

printf("input name, mail, tel, comp:\n");

scanf("%s %s %s %s",modi.name,modi.mail,modi.tel,modi.comp);

for(;head!=NULL && strcmp(head->name,name)!=0;head = head->next);

strcpy(head->name,modi.name);

strcpy(head->mail,modi.mail);

strcpy(head->comp,modi.comp);

strcpy(head->tel,modi.tel);

}

void delet(Node \*head){

char name[81];

printf("input name to delete:\n");

scanf("%s",name);

Node \*temp = head;

for(;head!=NULL && strcmp(head->name,name)!=0;head = head->next);

for(;temp->next!=head;temp = temp->next);

temp->next = head->next;

free(head);

}

void showall(Node \*head){

for(;head!=NULL;head = head->next)

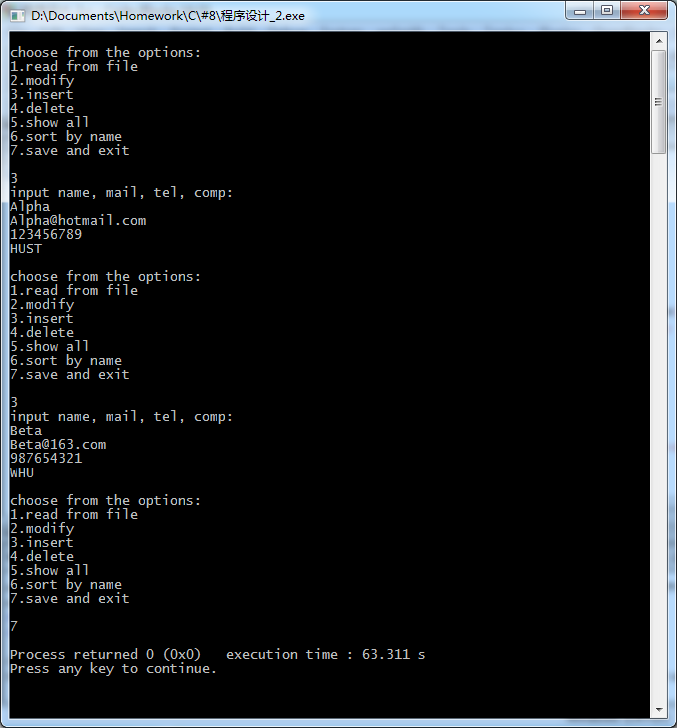
if((head->name)[0]!=0)

printf("%s\n%s\n%s\n%s\n\n",head->name,head->mail,head->tel,head->comp);

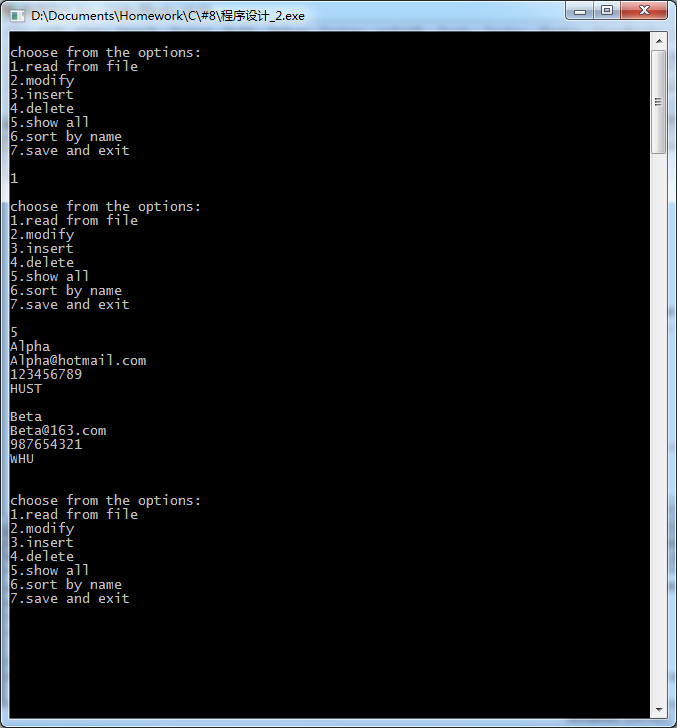
}

**【测试】**以下对基本功能进行测试：

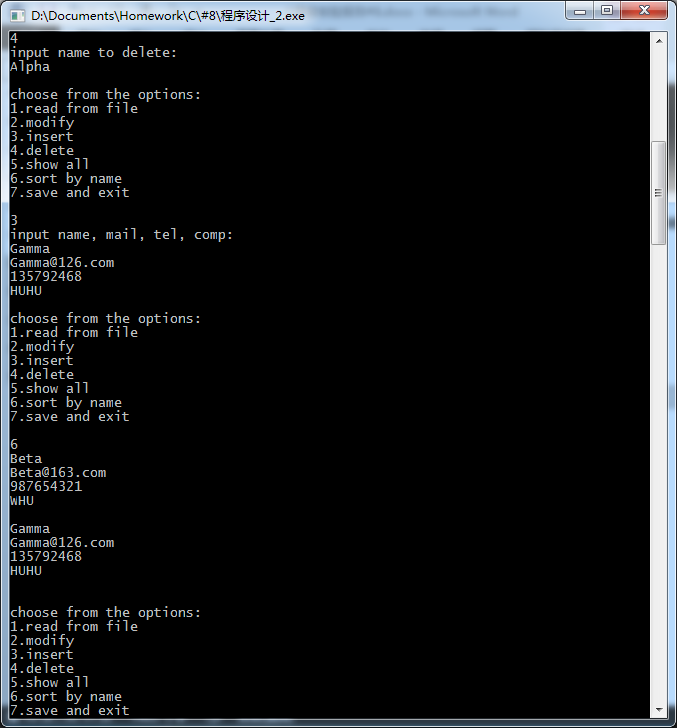
1. 录入两个人的数据并保存退出



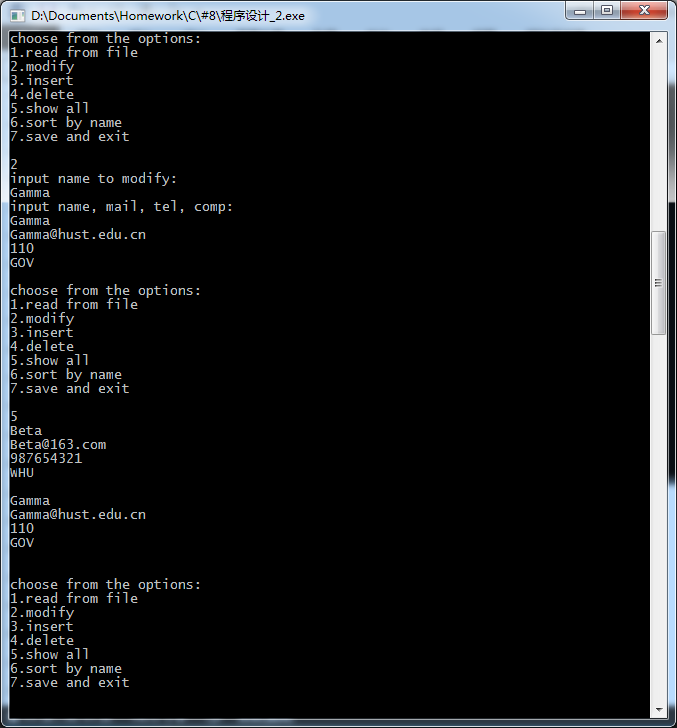
2．从文件中读取数据并显示



3．在2的基础上删除Alpha，插入Gamma，并排序



4．修改Gamma的电话和邮箱：



至此，所有功能此时完成，达到题目要求。

**8.3 实验小结**

通过本次实验，我熟悉了文本文件和二进制文件在磁盘中的存储方式，掌握了流式文件的读写方法，更重要的是，了解到了基本工程项目的构建方法，以及如何有效地依据要求对其进行优化。