# 实验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、实验内容

**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;

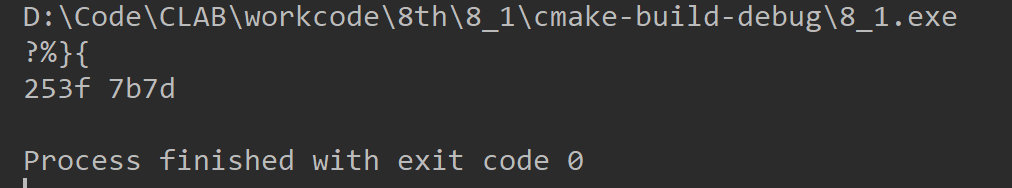
}

1. 请思考程序的输出结果，然后通过上机运行来加以验证。

第一行将输出十六进制25 3f 7b 7d ASCII码值对应的字符:?%}{

第二行将直接输出253f 7b7d

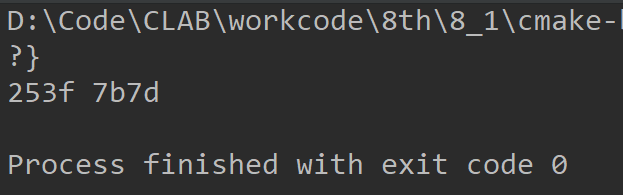
运行结果：



**图8-1 验证题的测试的运行结果 (1)**

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

运行结果：



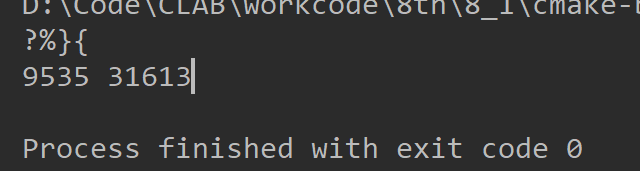
**图8-2 验证题的测试的运行结果 (2)**

第一行只打印出2个字符,为之前的第一和第三个

因为两次写入char长度的内容对应2个字符

1. 将fprintf(fp2,"%hx %hx",a,b) 改为 fprintf(fp2,"%d %d",a,b)结果有什么不同?

输出内容变成十进制。



**图8-3 验证题的测试的运行结果 (3)**

**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. #include<stdio.h>
2. #include<stdlib.h>
3. int main(int argc, char\* argv[])
4. {
5. char ch;
6. FILE \*fp;
7. if(argc!=2){
8. printf("Arguments error!\n");
9. exit(-1);
10. }
11. if((fp=fopen(argv[1],"r"))==NULL){ /\* fp 指向 filename \*/
12. printf("Can't open %s file!\n",argv[1]);
13. exit(-1);
14. }
15. while(ch=fgetc(fp)!=EOF) /\* 从filename中读字符 \*/
16. putchar(ch); /\* 向显示器中写字符 \*/
17. fclose(fp); /\* 关闭filename \*/
18. return 0;
19. }
20. 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。

错误：while(ch=fgetc(fp)!=EOF)

修改：while((ch=fgetc(fp))!=EOF)

修改后代码：

#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 错在此处\*/

putchar(ch);

/\* 向显示器中写字符 \*/

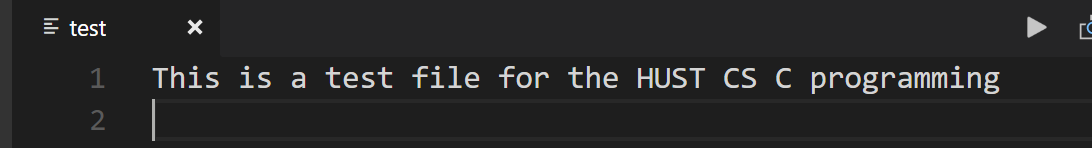
fclose(fp);

return 0;

}

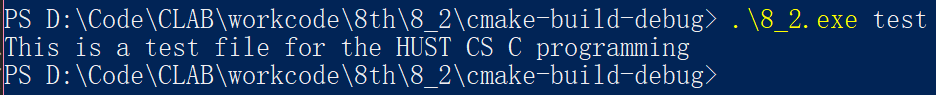
运行结果：

读入的文本文件为：



**图8-4 修改替换题读入文本截图**

结果为：



**图8-5 修改替换题（1）的测试运行结果**

1. 用输入输出重定向freopen改写上述源程序中的main函数。

程序代码：

#include<stdio.h>

#include<stdlib.h>

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

char ch;

if(argc!=2){

printf("Arguments error!\n");

exit(-1);

}

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

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

exit(-1);

}

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

putchar(ch);

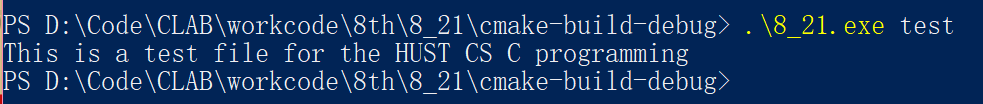
/\* 向显示器中写字符 \*/

fclose(stdout);

return 0;

}

运行结果：



**图8-6 修改替换题（2）的测试运行结果**

**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。

**解答：**

1. 算法思路：

利用

2）源程序清单

#include <stdio.h>

#include <stdlib.h>

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

freopen("test", "w", stdout);

if(argc<2){

fprintf(stderr, "arguments error!\n");

exit(-1);

}

for(int i = 1; i < argc; i++)

if(\*argv[i] >= 'a' && \*argv[i] <= 'z')

(\*argv[i]) -= 32;

for(int i = 1; i < argc; i++)

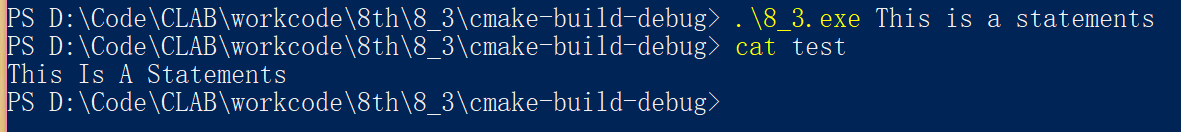
printf("%s ", argv[i]);

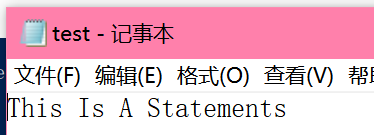
fclose(stdout);

return 0;

}

3）测试





**图8-8 编程题1的测试运行结果**

## 8.3 实验小结

此次实验让我学用了C语言中对文件进行操作的标准库函数。同时认识到了对文件进行处理的重要性与优劣之处。对文件进行操作是一个基本操作，属于必须学会的内容。经过这次实验，我了解到了基本的文件处理方法。

1. **验证题**

验证题展示了两种不同的文件类型在操作时的不同之处，经过解答验证题，我对C语言中两种文件类型的区别有了认识与了解。

1. **程序修改替换实验**

此题再次向我展示了基础之重要性。提醒我应对一些常用运算符的优先级以及结合性做到熟记。同时运用重定向函数的替换题，也使我对文件流这一概念有了更深刻的认识，对于C语言程序员来说，无论是键盘，屏幕，打印机，磁盘上的文件等输入输出设备或方式都被抽象成为文件，其抽象之后性质基本相同，因此可以利用重定向这一方式进行替换。这无疑带来了极大的便利，但是对其操作应该谨慎以防带来预料不到的错误。

1. **程序设计实验**

程序设计题中利用main函数参数读入字符串并处理单词首字母。同时继续熟悉文件的操作。