目录

[8 文件操作实验 1](#_Toc120122690)

[8.1 实验目的 1](#_Toc120122691)

[8.2 实验题目及要求 1](#_Toc120122692)

[8.3 实验小结 7](#_Toc120122693)

8 文件操作实验

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

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);

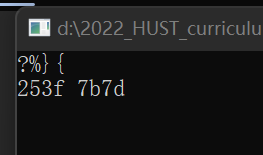
fclose(fp2);

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。
   1. 思考的结果

两行均为a和b的值。

* 1. 上机验证

可见，结果与预期并不相同

图8-1 实验题1-1截图

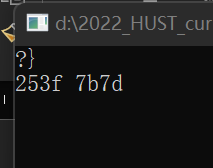
1. 将两处sizeof(short)均改为sizeof(char)结果有什么不同，为什么？
   1. 不同：

图8-2 实验题1-2截图

* 1. 原因：short 类型占两个字节的空间，而 char 占一个字节。所以只将a和b的第一个字节的数据写入了文件。

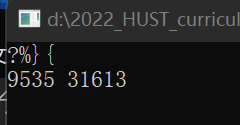
1. 将fprintf(fp2,"%hx %hx",a,b) 改为 fprintf(fp2,"%d %d",a,b)结果有什么不同。
   1. 不同之处：以十进制形式输出了a和b的值，而非十六进制
   2. 截图

图8-3 实验题1-3截图

**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。

01 #include<stdio.h>

02 #include<stdlib.h>

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

04 {

05 char ch;

06 FILE \*fp;

07 if(argc!=2){

08 printf("Arguments error!\n");

09 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

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

17 putchar(ch); /\* 向显示器中写字符 \*/

18 fclose(fp); /\* 关闭filename \*/

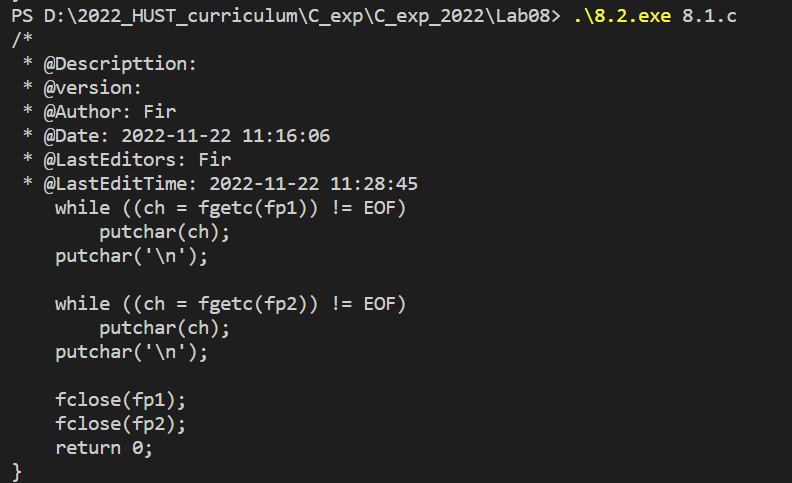
19 return 0;

20 }

1. 改错
   1. 第16行，运算符优先级出错，改为：

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

1. 运行截图

图8-4 实验题2-1截图

1. 用输入输出重定向freopen改写main函数。
   1. 源程序清单

#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 = freopen(argv[1], "r", stdin)) == NULL)

// if ((fp = fopen(argv[1], "r")) == NULL)

{ /\* fp 指向 filename \*/

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

exit(-1);

}

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

// 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;

}

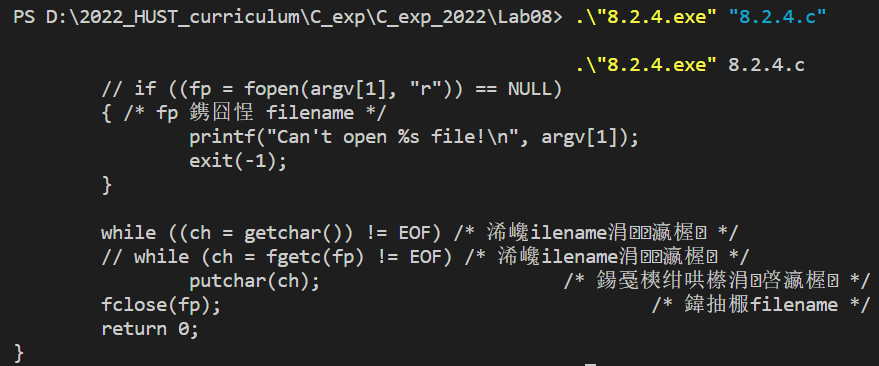
* 1. 运行截图

图8-4 实验题2-2截图

**3．程序设计**

（1）编写一个程序，用给定的字符串替换文件中的目标字符串，并显示输出替换的个数。

注意：读取的文件路径请使用experiment/src/step8/source.txt

若文件为

`There are moments in life when you miss someone so much that you just want to pick them from your dreams and hug them for real!`

样例输入：`you they`

样例输出：

`3`

`There are moments in life when they miss someone so much that they just want to pick them from theyr dreams and hug them for real!`

1. 思路
   1. 读入需要替换的字符串
   2. 打开文件，并从中读取内容。
   3. 从指定位置查找要替换的字串
      1. 储存查找位置之前的内容
      2. 储存要替换的字符串
      3. 将指针移动到指定位置
   4. 输出计数器
   5. 输出储存的内容——替换后的字符串
2. 源程序清单

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAXN 1000

int main()

{

// char filepath[] = "experiment/src/step8/source.txt";

char filepath[] = "source.txt";

FILE \*fp = fopen(filepath, "r");

char readin[MAXN] = {};

char output[MAXN]={};

// char replacement[MAXN]={};

char x[MAXN] = {}, y[MAXN] = {};

// gets(replacement);

scanf("%s%s", x, y);

int cnt = 0;

char \*p = NULL;

while (fgets(readin, MAXN, fp) != NULL)

{

p = readin; //当前在输出的位置

char \*end = p + strlen(readin) - 1;

char \*find = NULL;

while ((find = strstr(p, x)) != NULL)

{

for (; p < find && p <= end; p++)

{

// putchar(\*p);

sprintf(output+strlen(output),"%c",\*p);

}

// printf("%s", y);

sprintf(output+strlen(output),"%s",y);

cnt++;

p += strlen(x);

}

// printf("%s", p);

sprintf(output+strlen(output),"%s",p);

memset(readin, 0, MAXN \* sizeof(char));

}

printf("%d\n", cnt);

printf("%s",output);

fclose(fp);

}

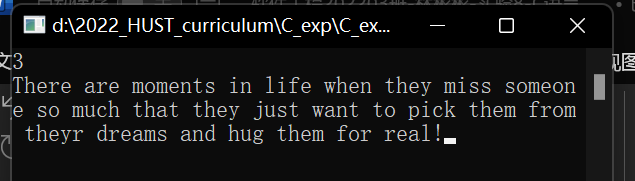
1. 运行截图

图8-5 程序设计题截图

8.3 实验小结

（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。