

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级： 软件工程2003班**

**学 号： U202010851**

**姓 名： 侯皓斐**

**指导教师： 唐赫**

**报告日期： 2020.12.22**

**软件工程**

**目 录**

[**8 文件操作实验 1**](#_Toc404837920)

[8.1 实验目的 1](#_Toc404837921)

[8.2 实验内容 1](#_Toc404837922)

8[.3 实验小结 20](#_Toc404837923)

# 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。

**解答：**

文件类型Fp1按照二进制位输出，低字节在先，读入时按照字节读入，按照ASCII码对应3f -> ? 25->% 7d->} 7b->{，其输出结果应该为?%}{

而文件类型fp2按照字符读写，输出时直接输出其十六进制数的字符串，读入时读入字符串并输出，其结果应该为253f 7b7d。

测试结果如下：



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

**解答：**

文件类型Fp1按照二进制位输出，低字节在先，但是short是两个字节，char是一个字节，其故只输出了低字节。读入时按照字节读入，按照ASCII码对应3f -> ? 7d->}，其输出结果应该为?}

测试结果如下：



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

**解答：**

文件类型fp2按照字符读写，输出时直接输出其十进制数的字符串，读入时读入字符串并输出，其结果应该为9535 31613。



**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

#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;

}

（1）源程序中存在什么样的错误（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。

**解答：**

1. 错误修改：

通过运行程序发现，程序输出乱码空格字符串。逐步调试发现，ch结果永远为1.故发现while括号内语句存在问题。

1) 第16行因为!=运算优先级高于赋值运算，若ch始终只能是布尔值0或1，正确形式为：

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

1. 错误修改后运行结果：

参数设置为本程序”2.c”。

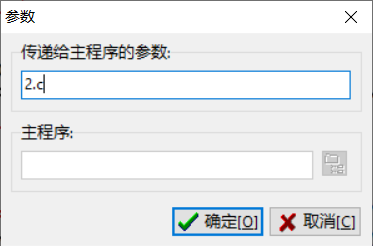


图8-1 源程序修改替换题的参数

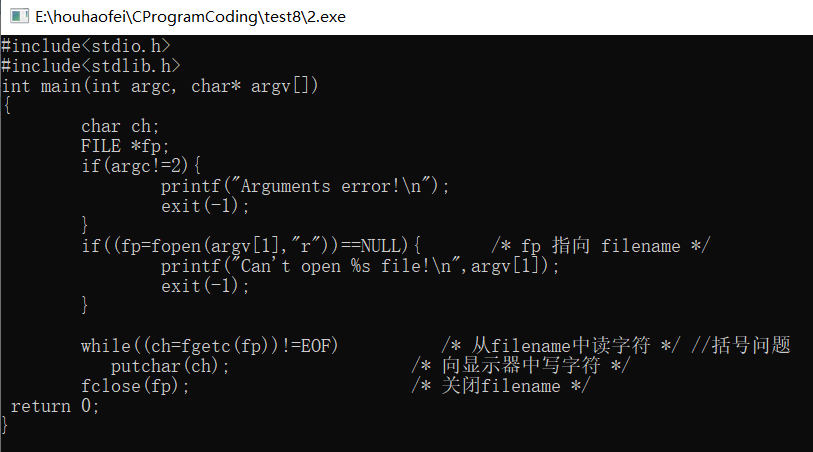


图8-2 源程序修改替换题的运行结果

（2）用输入输出重定向freopen改写main函数。

**解答：**

整体代码框架基本保持不变。

freopen()函数用于文件流的的重定向，一般是将 stdin、stdout 和 stderr 重定向到文件。

所谓重定向，就是改变文件流的源头或目的地。stdout（标准输出流）的目的地是显示器，printf()是将流中的内容输出到显示器；可以通过freopen()将stdout 的目的地改为一个文件（如output.txt），再调用 printf()，就会将内容输出到这个文件里面，而不是显示器。

freopen()函数的原型为：

FILE \*freopen(char \*filename, char \*type, FILE \*stream);

filename为要重定向到的文件名；type为文件打开方式， stream为被重定向的文件流（一般是 stdin、stdout 或 stderr）。成功则返回指向filename文件的指针，否则返回NULL。

代码实现如下：

#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 \*/

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

exit(-1);

}

while(scanf("%c", &ch)!=EOF) /\* 从filename中读字符 \*/ //括号问题

putchar(ch); /\* 向显示器中写字符 \*/

fclose(stdout); /\* 关闭filename \*/

return 0;

}

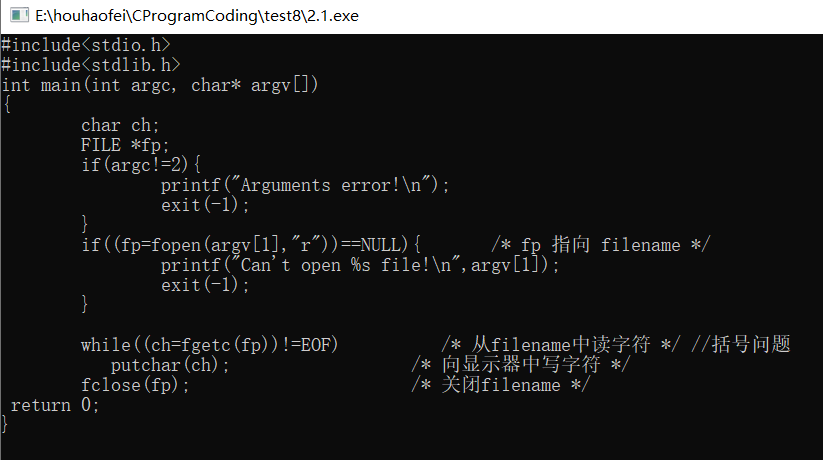


图8-3 源程序修改替换题（2）的运行结果

**7.2.3 程序设计**

（1） 本关任务：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

**解答：**

1. 解题思路：

输出流不需要修改。故可以使用分件类型打开该文件，读取所需修改的文件。利用fgets函数循环将文件内容全部读入（而fgets函数如果发生错误，返回一个空指针）。

而此处查找替换字符串可以使用暴力查找匹配。其时间复杂度为。也可以使用KMP算法查找。但此处已经知道文件不大，故设计算法流程图如下：

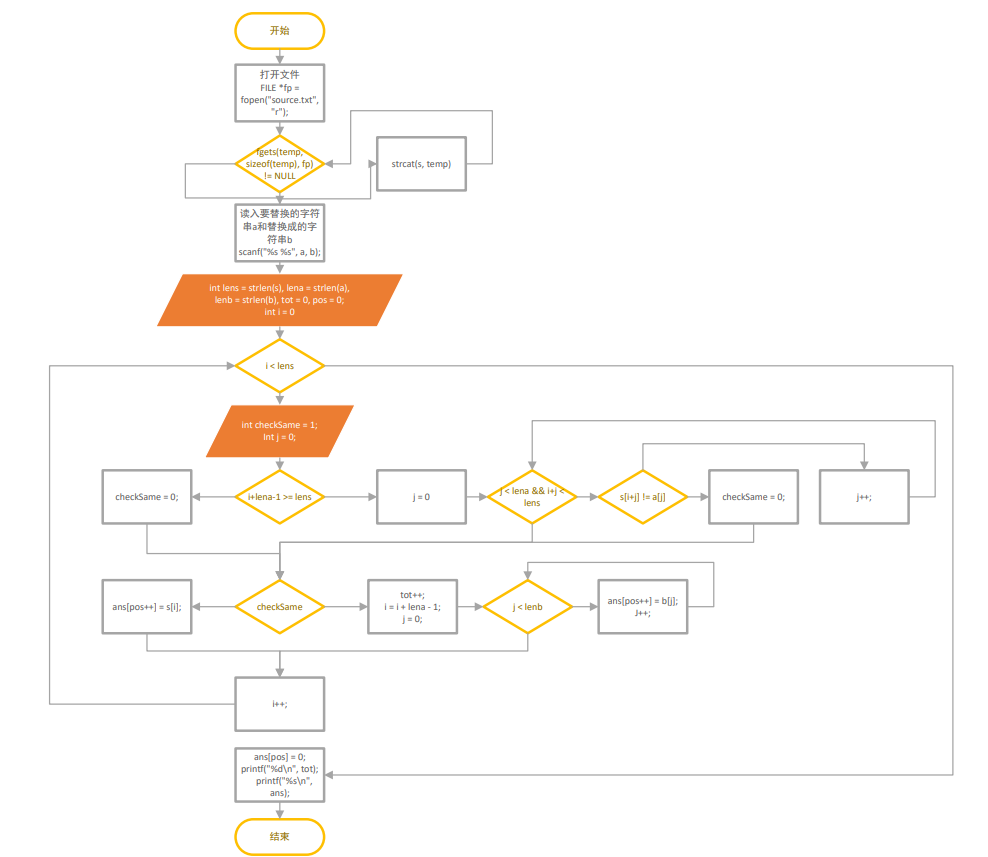


图8-4程序设计题1的算法流程示意图

1. 源程序清单

#include <stdio.h>

#include <string.h>

char s[100000], temp[10000], ans[100000];char a[1000], b[1000];

int main () {

FILE \*fp = fopen("source.txt", "r");

s[0] = '\0';

while(fgets(temp, sizeof(temp), fp) != NULL)

strcat(s, temp);

scanf("%s %s", a, b);

int lens = strlen(s), lena = strlen(a), lenb = strlen(b), tot = 0, pos = 0;

for(int i = 0; i < lens; i++) {

int checkSame = 1;

if(i+lena-1 >= lens)

checkSame = 0;

else

for(int j = 0; j < lena && i+j < lens; j++)

if(s[i+j] != a[j]) {

checkSame = 0;

break;

}

if(checkSame) {

tot++;

i = i + lena - 1;

for(int j = 0; j < lenb; j++)

ans[pos++] = b[j];

}

else

ans[pos++] = s[i];

}

ans[pos] = 0;

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

printf("%s\n", ans);

return 0;

}

3）测试

（a） 测试数据：

测试文章source.txt如下：

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!

I love you! Do you love me?

根据题目要求以及各种可能出现的错误情况构造数据如表8-1所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 测试  用例 | 输 入 | 理 论 结 果 | 运 行 结 果 |
| 被替换串与替换串 |
| 例1 | you they | 5  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!  I love they! Do they love me? | 5  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!  I love they! Do they love me? |
| 例2 | want love | 1  There are moments in life when you miss someone so much that you just love to pick them from your dreams and hug them for real!  I love you! Do you love me? | 1  There are moments in life when you miss someone so much that you just love to pick them from your dreams and hug them for real!  I love you! Do you love me? |
| 例3 | work w | 0  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!  I love you! Do you love me? | 0  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!  I love you! Do you love me? |

（b） 对应测试测试用例1的运行结果如图8-5所示。

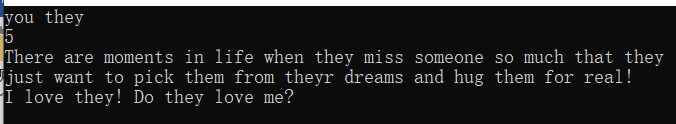


图8-5 编程题1的测试用例一的运行结果

对应测试测试用例2的运行结果如图8-6所示。

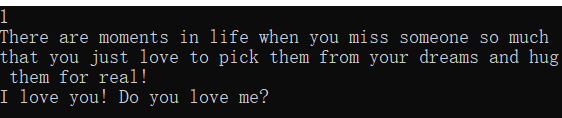


图8-6 编程题1的测试用例二的运行结果

对应测试测试用例3的运行结果如图8-7所示。

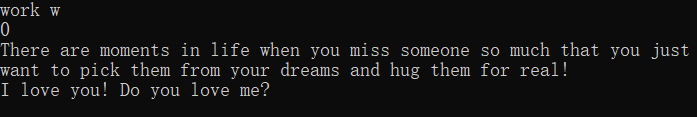


图8-7 编程题1的测试用例三的运行结果

说明上述的运行结果与理论分析吻合，验证了程序的正确性。

## 8.3 实验小结

主要叙述实验过程中遇到的问题，如何解决的，通过分析、结果问题后的体会。

通过本次实验，我较深入地学习了C语言的文件操作，特别是对二进制文件的读写掌握的更加深刻。先前的数据结构课程设计中，由于文件读入的原因出了不少差错，进了不少的坑，现在通过文件类型的程序验证题从理论上更深入的理解了二进制文件的读写。也通过源程序修改替换题和程序设计题1掌握了文件类型，和输入输出重定向进行文件读写的实现方式。也在程序设计题1中使用文件读写解决了一定的实际问题。