实验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。
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。

解答：

1. 输出结果：

?%}{

253f 7b7d

1. 输出结果变为：

?}

253f 7b7d

因为在写入文件fp1时，a与b只取了一个字符大小的前一部分（Windows是大端法系统，高字节储存在低地址中）。

1. 输出结果变为：

?%}{

9535 31613

同（2）因为写入文件fp2时，a与b按整数形式写入，而读取时按字符读取。

**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;

}

解答：

只需将while(ch=fgetc(fp)!=EOF)改为while((ch=fgetc(fp))!=EOF)即可

1. 用输入输出重定向freopen改写main函数。

解答：

改写后的main函数如下：

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

{

FILE \*fp;

char ch;

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

if (argc != 2)

{

printf("Arguments error!\n");

exit(-1);

}

if (fp == NULL)

{

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

exit(-1);

}

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

putchar(ch);

}

fclose(fp)

return 0;

}

**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!`

解答：

1. 算法流程图如图3-1所示：

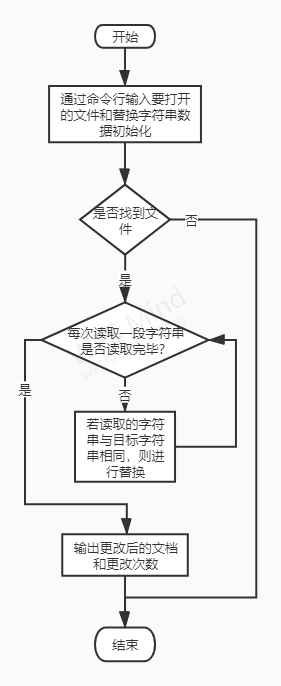


图3-1 程序设计（1）流程图

1. 源程序清单：

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

int count = 0, flag = 0;

FILE \*fp, \*new;

fpos\_t position;

fp = fopen("D:/abc.txt", "r");

new = fopen("D:/abc1.txt", "w+");

char target[10], alter[10], tempc;

scanf("%s%s", target, alter);

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

{

if (tempc == target[0])

{

for (int i = 1; i < strlen(target); i++)

{

if (i == 1)

{

fgetpos(fp, &position);

}

if ((tempc = fgetc(fp)) == target[i])

{

flag++;

}

else

{

break;

}

}

if (flag == strlen(target) - 1)

{

fprintf(new, "%s", alter);

count++;

}

else

{

fsetpos(fp, &position);

fprintf(new, "%c", target[0]);

}

}

else

{

fprintf(new, "%c", tempc);

}

flag = 0;

}

fclose(fp);

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

rewind(new);

while ((tempc = fgetc(new)) != EOF)

{

putchar(tempc);

}

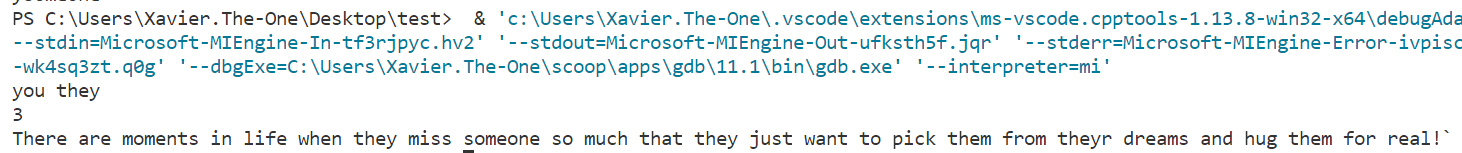
fclose(new);

remove("D:/abc.txt");

rename("D:/abc1.txt","D:/abc.txt");

}

1. 测试：
2. 测试用例如题干所示
3. 对应测试用例的运行结果如图3-2所示：



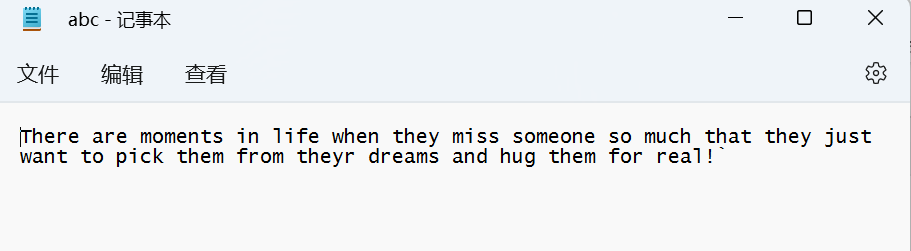


图3-2 程序设计（1）运行结果

8.2实验小结

本次实验主要学习了C语言中流式文件的使用和文件的存储方式。文件在磁盘中以二进制或字符形式存储，在程序中可以通过函数以特定的格式打开或读写。格式决定了程序能对文件进行的操作。

通过实验，我还认识到了虽然文件指针地址不随fgetc函数的使用而变化，但文件内部其实存在fpos\_t类型的文件内部指针，决定了当前fgetc函数会读取哪一位置的字符。可以通过fgetpos与fsetpos函数进行对文件内部指针的操作。

借由文件，程序可以进行更广泛的数据读取，不再局限于键盘的输入。流式文件的使用尽管复杂，但更贴近底层，固定的打开格式也在一定程度上保护了文件的读写安全。