

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级：**

**学 号：**

**姓 名：**

**指导教师： 余林琛**

**报告日期：**

**网络空间安全学院**

**目□□录**（黑体小2号加粗居中这里□代表空格）

[**1□□□表达式和标准输入输出实验 1**](#_Toc404837920)

[1.1□□实验目的 1](#_Toc404837921)

[1.2□□实验内容 1](#_Toc404837922)

[1.3□□实验小结 1](#_Toc404837923)

[**2□□□流程控制实验 2**](#_Toc404837924)

[2.1□□实验目的 1](#_Toc404837921)

[2.2□□实验内容 1](#_Toc404837922)

[2.3□□实验小结 1](#_Toc404837923)

[**3□□□函数与程序结构实验 3**](#_Toc404837929)

[3.1□□实验目的 1](#_Toc404837921)

[3.2□□实验内容 1](#_Toc404837922)

[3.3□□实验小结 1](#_Toc404837923)

[**4□□□编译预处理实验 4**](#_Toc404837934)

[4.1□□实验目的 1](#_Toc404837921)

[4.2□□实验内容 1](#_Toc404837922)

[4.3□□实验小结 1](#_Toc404837923)

[**5□□□数组实验 5**](#_Toc404837938)

[5.1□□实验目的 1](#_Toc404837921)

[5.2□□实验内容 1](#_Toc404837922)

[5.3□□实验小结 1](#_Toc404837923)

[**6□□□指针实验 6**](#_Toc404837943)

[6.1□□实验目的 1](#_Toc404837921)

[6.2□□实验内容 1](#_Toc404837922)

[6.3□□实验小结 1](#_Toc404837923)

[**7□□□结构与联合实验 7**](#_Toc404837948)

[7.1□□实验目的 1](#_Toc404837921)

[7.2□□实验内容 1](#_Toc404837922)

[7.3□□实验小结 1](#_Toc404837923)

[**8□□□文件实验 8**](#_Toc404837953)

[8.1□□实验目的 1](#_Toc404837921)

[8.2□□实验内容 1](#_Toc404837922)

[8.3□□实验小结 1](#_Toc404837923)

[**参考文献 9**](#_Toc404837957)

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

8.2.1．文件类型的程序验证题

设有程序：

#include <stdio.h>

#include<stdlib.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）？（文本为3f）}（文本为7D）

253f 7d7d

（2）?(3f) %(25) }(7d) {(7b)

（3）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

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 \*/

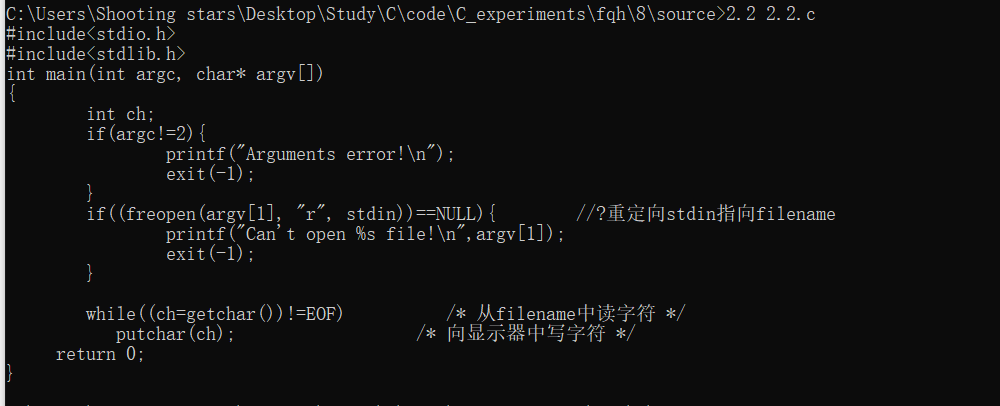
return 0;

}

（2）用输入输出重定向freopen改写main函数。

**解答：**

1. 错误修改：
2. 第5行ch应定义为int
3. 第16行应该为while((ch=fgetc(fp))!=EOF)
4. 错误修改后运行结果：



**2-2-1运行结果示意图**

**2．解答：**

代码如下：

#include<stdio.h>

#include<stdlib.h>

int main(int argc, char\* argv[])

{

    int ch;

    if(argc!=2){

        printf("Arguments error!\n");

        exit(-1);

    }

    if((freopen(argv[1], "r", stdin))==NULL){        //?重定向stdin指向filename

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

        exit(-1);

    }

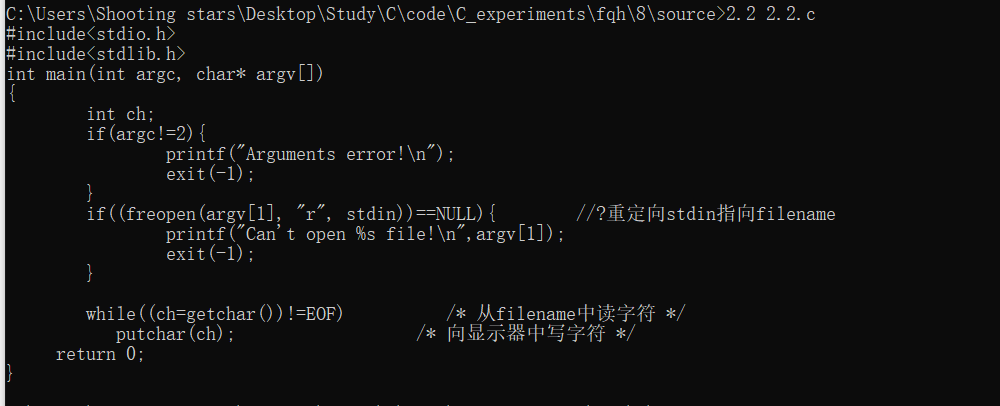
    while((ch=getchar())!=EOF)          /\* 从filename中读字符 \*/

       putchar(ch);                  /\* 向显示器中写字符 \*/

     return 0;

}

运行结果：



**2-2-2运行结果示意图**

**8.3程序设计题**

**8.3.1**编写一个程序replace，采用命令行方式，用给定的字符串替换指定文件中的目标字符串，并显示输出替换的个数。例如，命令行：

replace filename.txt you they

you like your self

**解答：**

流程图：



**图8-3-1 程序设计流程图**

代码如下：

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int replace(char \*,char \*,char \*, char \*);

int main(int argc, char \*argv[])

{

    //判断参数是否正确

    if(argc != 4)

    {

        printf("Arguments error!\n");

        return 0;

    }

    //定义变量与打开文件

    FILE \*fp1,\*fp2;

    char buff[255],temp[255], \*p;

    int len = strlen(argv[2]),n = 0;

    if((fp1=fopen(argv[1], "r")) == NULL)

    {

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

        return 0;

    }

    fp2 = fopen("temp.txt", "w");

    //replace操作

    while(fgets(buff, 255, fp1))

    {

        p = buff;

        strcpy(temp,"");

        while(\*p != '\0')

        {

            if(replace(p,temp,argv[2], argv[3]))

            {

                p = p+len;

                n++;

            }

            else

                p++;

        }

        fputs(temp, fp2);

    }

    freopen(argv[1], "w",fp1 );

    freopen("temp.txt", "r",fp2);

    while(fgets(buff, 255, fp2))

    fputs(buff, fp1);

    fclose(fp1);

    fclose(fp2);

    remove("temp.txt");

    printf("total:%d\n", n);

    return 0;

}

int replace(char \*p,char\*temp, char \*t1, char \*t2)

{

    int flag = 0;

    char \*p0 = p;

    while(\*temp++);

    temp--;

    while(\*p++ == \*t1++);

    t1--;

    if(\*t1 == '\0')

    {

        while(\*temp++ = \*t2++);

        flag = 1;

    }

    else

    {

        \*temp++ = \*p0;

        \*temp = '\0';

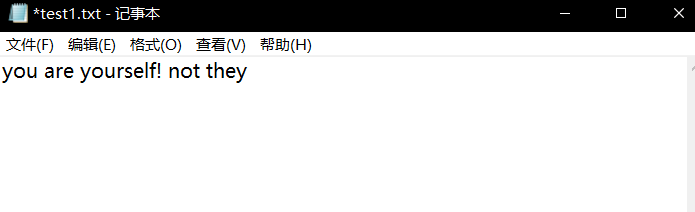
    }

    return flag;

}

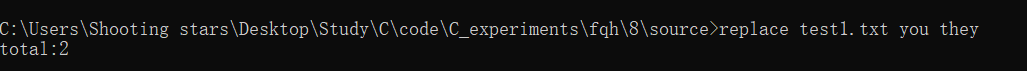
运行结果：

替换前



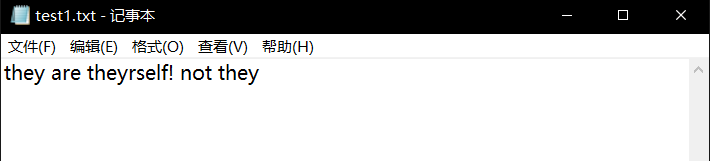
**图8-3-2运行结果示意图**

运行结果：



**图8-3-3运行结果示意图**

替换后：



**图8-3-4运行结果示意图**

（2）从键盘输入10个单精度浮点数，以二进制形式存入文件float.dat中。再从文件中读出这10个单精度浮点数显示在屏幕上。之后要求将float.dat中的单精度浮点数按字节读出来，观察写入文件的浮点数字节数据是不是和计算机内存中表示的浮点数字节数据一致。

**解答：**

流程图：

**图8-3-5运行结果示意图**

代码如下：

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define N 10

#define M (N\*4)

int main()

{

    FILE \*fp;

    float a[N],b[N];

    char c[M], \*p;

    int i,t;

    for(i = 0; i<N; i++)

        scanf("%f", &a[i]);

    if(!(fp = fopen("float.dat", "wb")))

    {

        printf("Can't open float.dat file !\n");

        return 0;

    }

    fwrite(a, sizeof(float), N, fp);

    freopen("float.dat", "rb", fp);

    fread(b,sizeof(float), N, fp);

    rewind(fp);

    fread(c, sizeof(char), M, fp);

    for(i = 0; i<N; i++)

    printf("%f\n", b[i]);

    printf("字节比较:");

    for(i = 0; i<N; i++)

    {

        t = 4\*i;

        p = (char\*) &a[i];

        printf("第%d个小数的字节比较：\n", i);

        printf("第1个字节:\n");

        printf("源文件中:%x\t", \*p++ & 0xff);

        printf("二进制文件中:%x\n", c[t++] & 0xff);

        printf("第2个字节:\n");

        printf("源文件中:%x\t", \*p++ &0xff);

        printf("二进制文件中:%x\n", c[t++] &0xff);

        printf("第3个字节:\n");

        printf("源文件中:%x\t", \*p++ &0xff);

        printf("二进制文件中:%x\n", c[t++] &0xff);

        printf("第4个字节:\n");

        printf("源文件中:%x\t", \*p++ & 0xff);

        printf("二进制文件中:%x\n", c[t++] &0xff);

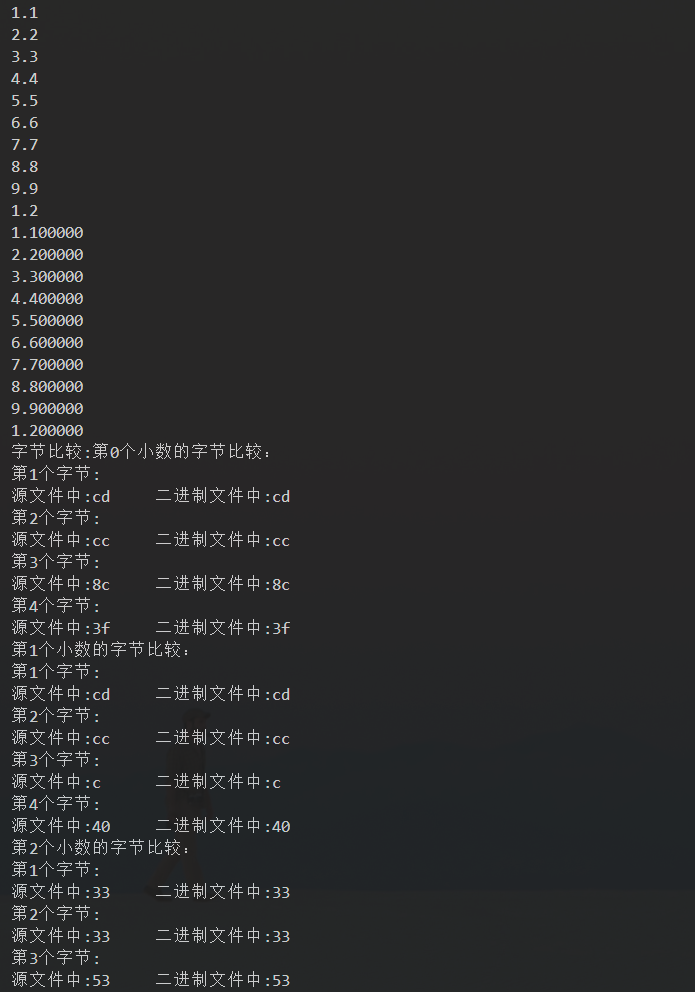
    }

    fclose(fp);

    return 0;

}

运行结果：



**图8-3-6运行结果示意图**

## 1.3 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。

在本次实验中遇到了大量有关文件的函数，只有熟悉各个函数的含义与用法，并多次使用，才能灵活掌握文件的用法，将其应用于实践中

参考文献

[1] 曹计昌,卢萍,李开. C语言程序设计,北京： 科学出版社,2013

[2] 李开,卢萍,曹计昌. C语言实验与课程设计, 北京：科学出版社,2011