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>

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。

（下面只考虑小端序机器上的运行情况）

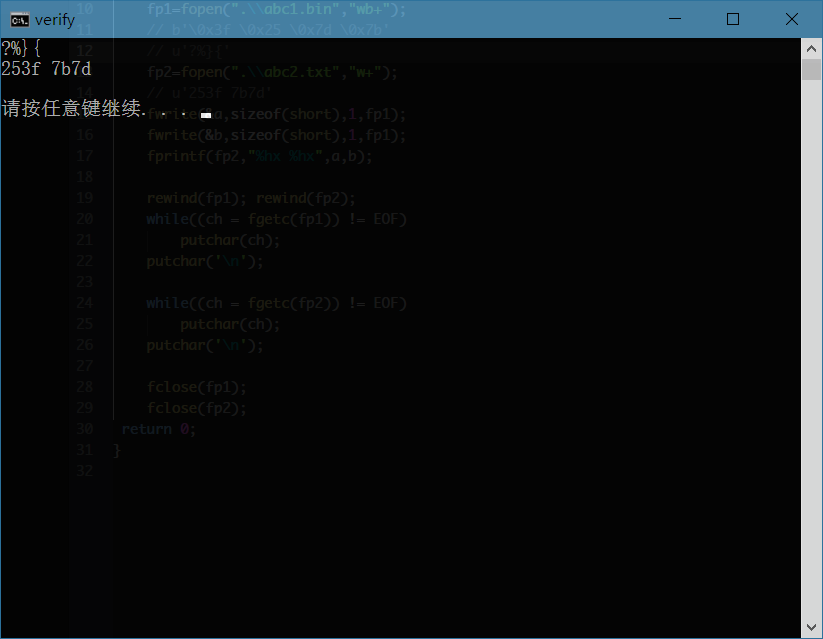
（考虑到程序兼容性，在编译时已将文件目录均改为当前目录）

**预期输出结果**

?%}{

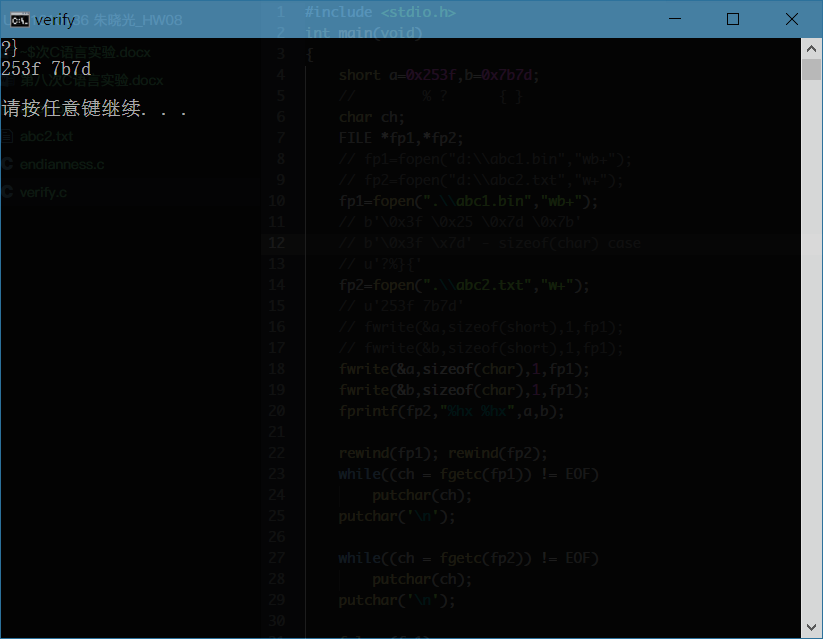
253f 7b7d

**实际运行结果截图**



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

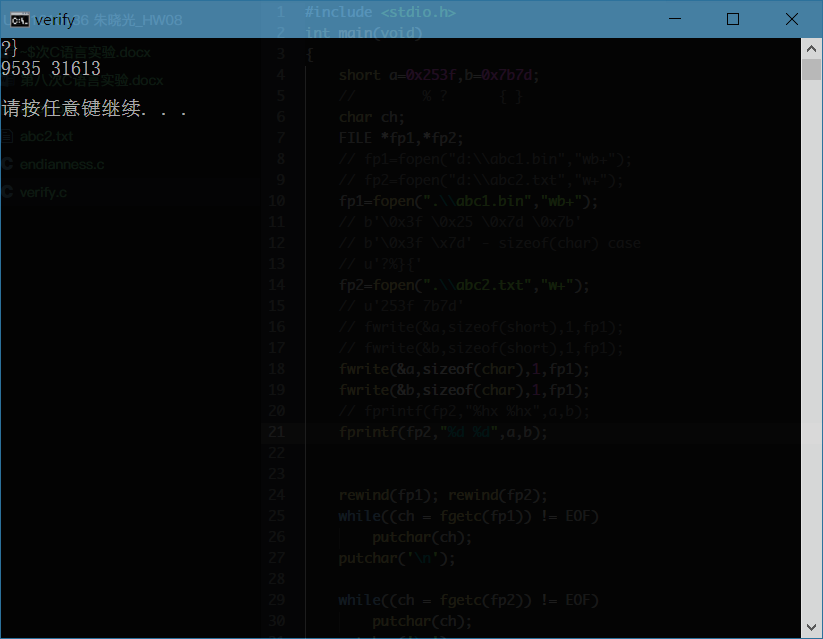
输出结果中第一行变为：?}



原因：fwrite()函数的参数限制最多能够写进的单位数量为1。因此更改写入时解释数据用的单位大小后，多余的位数被舍去，只留下低位。

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

输出结果的第二行变为：9535 31613



8.2.2 源程序修改替换题

1. 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

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 \*/

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

putchar(ch); /\* 向显示器中写字符 \*/

fclose(fp); /\* 关闭filename \*/

return 0;

}

**运行结果截图**



1. 用输入输出重定向freopen改写上述源程序中的main函数。

对上一题中第11行做如下修改：

if((fp=freopen(argv[1], ”r”, stdin)) == NULL) {

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。

**源程序清单**

#include <stdio.h>

#include <stdlib.h>

#define isLowerCase(c) ( ((c)<='z' && (c)>='a') ? 1 : 0 )

void lowerToUpper(char \*c) { \*c = \*c + 'A' - 'a'; }

void capitalize(char \*str) {

if (!str) { return; }

if (isLowerCase(\*str)) { lowerToUpper(str); }

for (++str; \*str; ++str) {

if (\*(str-1)==' ' && isLowerCase(\*str)) { lowerToUpper(str); }

}

return;

}

void main(void) {

char str[81] = {0};

char \*buf = str;

for (; (\*buf=getchar()) != '\n'; ++buf) ;

\*buf = 0;

capitalize(str);

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

FILE \*fp;

#ifdef \_\_WIN32\_\_

fp = fopen(".\\test", "w+");

#elif linux

fp = fopen("./test", "w+");

#else

puts("OS type not recognized!");

exit(-1);

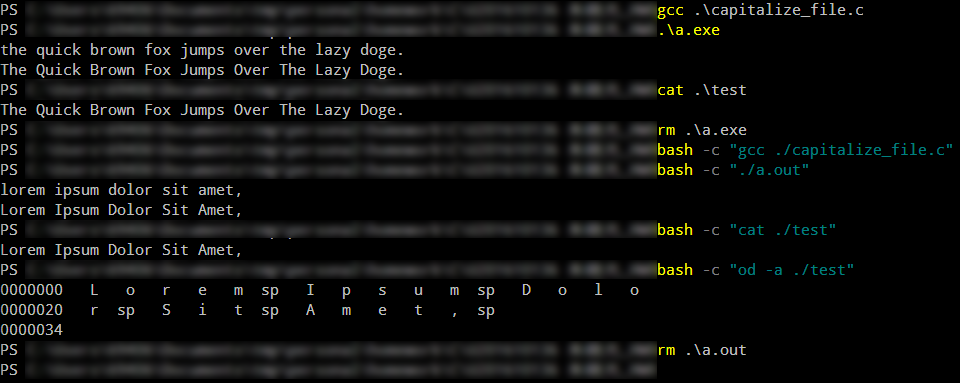
#endif

fputs(str, fp);

fclose(fp);

}

**运行结果截图**



8.3 实验小结

通过这次实验，我熟悉了C语言中基本的文件操作，但使用还不是很熟练，有些概念还不是太清楚，需要在课后加强。

另外，我也尝试编写了通过宏定义来同时适配Windows和\*nix环境的程序（编程设计题第一小题）。