

第14讲 文件操作

本讲主要内容

- 文件的概念
- 文件类型指针
- 文件的打开与关闭
- 文件的读写

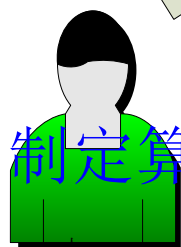


问题： $x=10$, $y=x+3$, $y=?$

解题思路：

第n步：设整型变量a,b; $a=10$

第n+1步：将b与3相加，赋给a；



制定算法

成功后的享受

C程序代码/运行结果



计算机

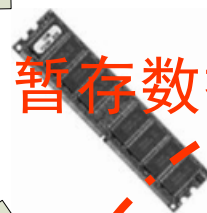
输出

程序执行过程



输入

C程序
`int a,b; a=10;
a=b+3;`



暂存数据

计算



执行指令

二进制指令：01101110

二进制文件：huang.c

磁盘读写



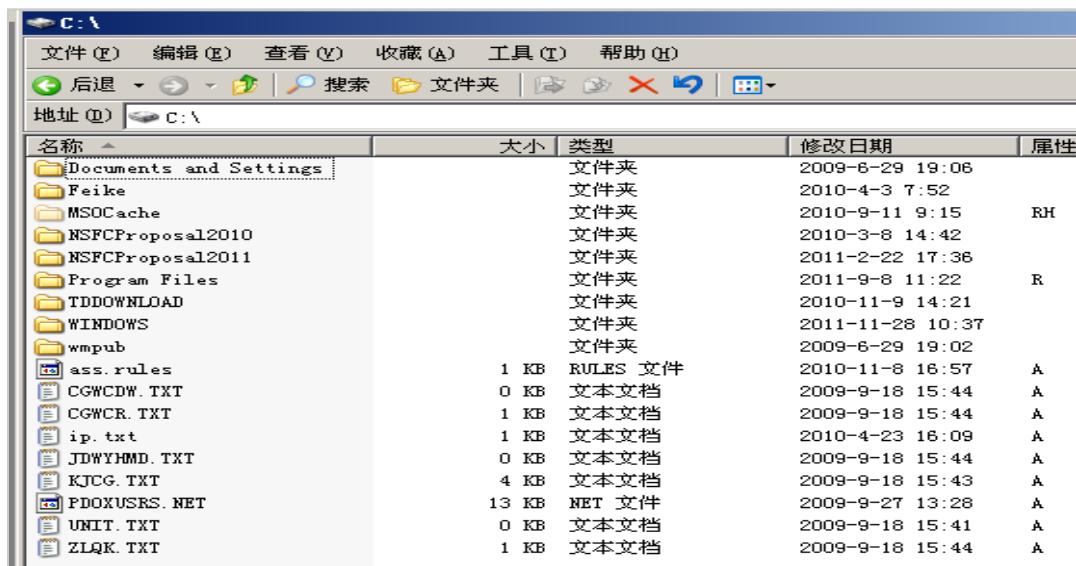
保存程序

14.1 文件的概念

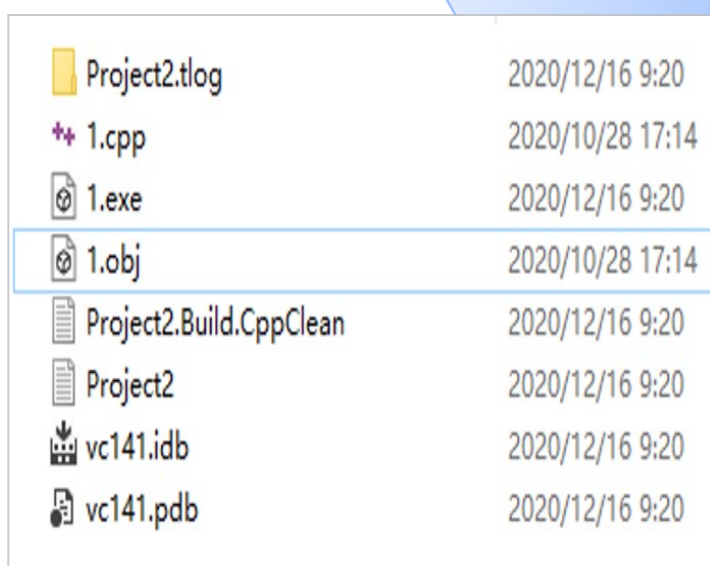
□ 文件(file)：记录在磁盘上数据集合，以文件名作为访问标识

□ 类型

- ✓ 按文件存储介质分：磁盘文件、磁带文件、U盘文件等；
- ✓ 按文件内容区分：源程序文件、目标文件、数据文件等；
- ✓ 按数据编码格式分：ASCII码文件和二进制文件。

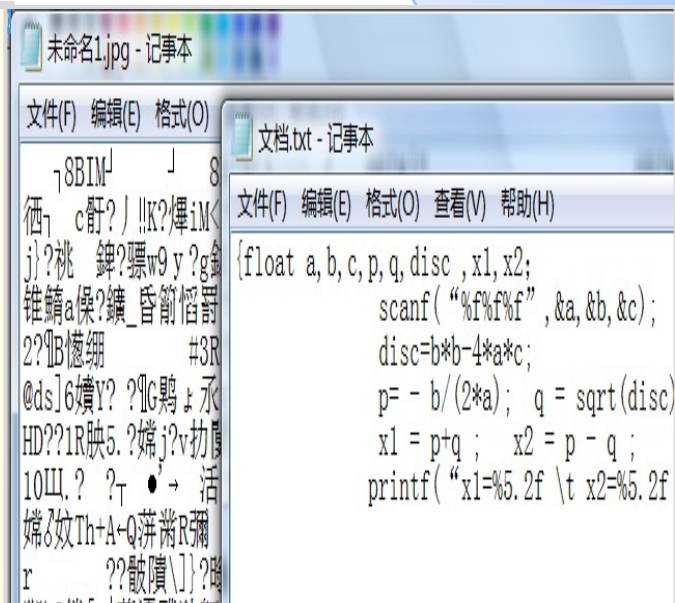


名称	大小	类型	修改日期	属性
Documents and Settings		文件夹	2009-6-29 19:06	
Feike		文件夹	2010-4-3 7:52	
MSOCache		文件夹	2010-9-11 9:15	RH
NSFCProposal2010		文件夹	2010-3-8 14:42	
NSFCProposal2011		文件夹	2011-2-22 17:36	
Program Files		文件夹	2011-9-8 11:22	R
TDDOWNLOAD		文件夹	2010-11-9 14:21	
WINDOWS		文件夹	2011-11-28 10:37	
wmpub		文件夹	2009-6-29 19:02	
ass.rules	1 KB	RULES 文件	2010-11-8 16:57	A
CGWCDW.TXT	0 KB	文本文档	2009-9-18 15:44	A
CGWCR.TXT	1 KB	文本文档	2009-9-18 15:44	A
ip.txt	1 KB	文本文档	2010-4-23 16:09	A
JDWYHMD.TXT	0 KB	文本文档	2009-9-18 15:44	A
KJCG.TXT	4 KB	文本文档	2009-9-18 15:43	A
PDOXUSRS.NET	13 KB	NET 文件	2009-9-27 13:28	A
UNIT.TXT	0 KB	文本文档	2009-9-18 15:41	A
ZLQK.TXT	1 KB	文本文档	2009-9-18 15:44	A

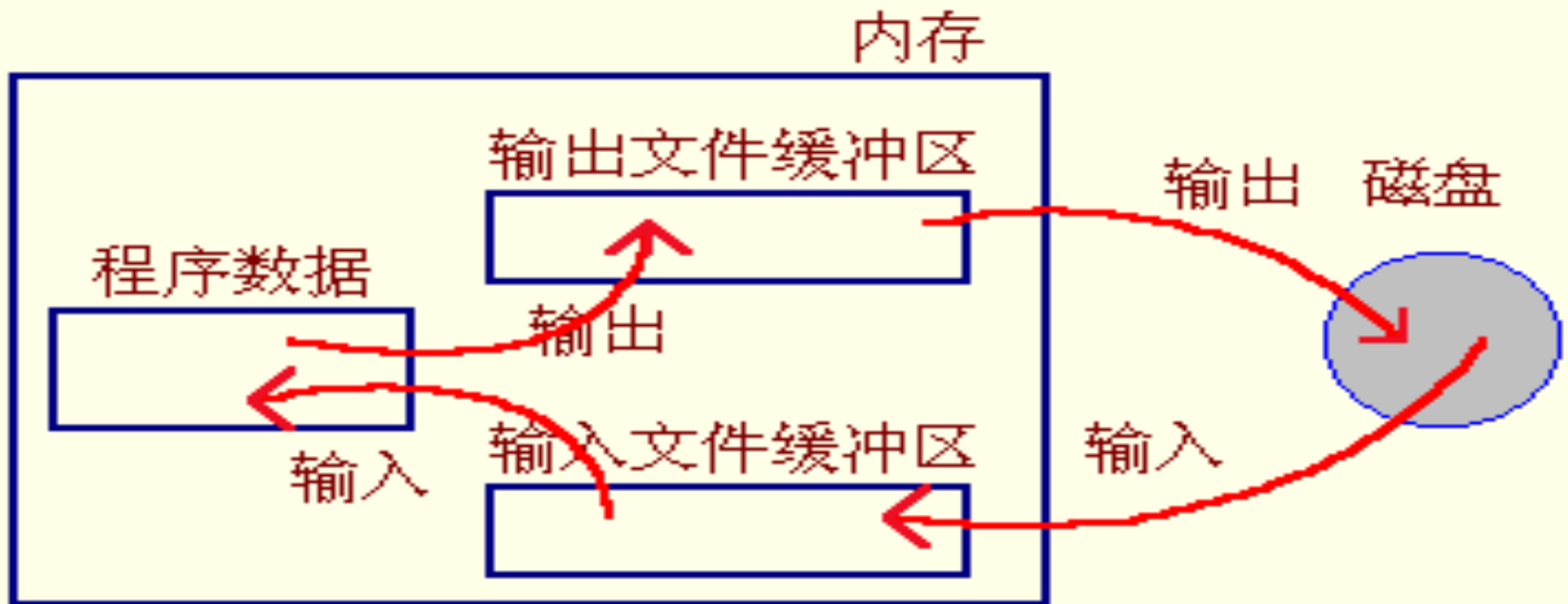


Project2.tlog	2020/12/16 9:20
1.cpp	2020/10/28 17:14
1.exe	2020/12/16 9:20
1.obj	2020/10/28 17:14
Project2.Build.CppClean	2020/12/16 9:20
Project2	2020/12/16 9:20
vc141.idb	2020/12/16 9:20
vc141.pdb	2020/12/16 9:20

- ❑ ASCII码文件: 又称为“文本文件”(text)。文件中每一个字节存放一个ASCII字符。优点: 可编辑性好
- ❑ 二进制文件: 又称“内存映像”文件。文件中每个数据格式与其在内存中存储结构相同。优点: 节省存储空间
- ❑ 问题: (1) 数据在内存中存储格式是啥?
- ❑ (2) 数据在屏幕显示格式是啥?
- ❑ (3) 哪种文件的读写速度快? 为啥?



- 缓冲文件系统：操作系统在内存中为每一个正在使用的文件开辟一个读写缓冲区，又称为高级文件系统；
- 非缓冲文件系统：操作系统不开辟读写缓冲区。由用户来开辟，称为低级文件系统。



§ 14.2 文件类型指针

- 应用程序调用缓存文件系统中的文件时，需要以下信息：
 - ✓ 与该文件对应的内存缓冲区的地址；
 - ✓ 缓冲区中未被处理的字符串；
 - ✓ 文件操作方式；
 - ✓ 文件当前的读写位置。

□ 缓冲文件系统会为每个文件设立一个“文件信息区”，在 [stdio.h](#) 中，定义为 [FILE](#) 类型数据

```
typedef struct
{ int _fd;           //文件号
  int _cleft; // 缓冲区中剩下的字节数
  int _mode;    // 文件操作模式
  char * _nextc; //下一个字节的位置
  char \* \_buff; //文件缓冲区位置
} FILE;
```

§ 14.3 文件的打开与关闭

❑ 磁盘文件操作的“3部曲”

- ✓ 先打开
- ✓ 后读写
- ✓ 最后关闭

❑ 打开文件：以某种方式从磁盘上查找或创建指定文件，并获取该文件的FILE类型数据

`FILE *fopen (const char *filename, const char *mode;)`

❑ 其中，filename：文件名；mode：打开方式见下表，FILE *：返回值

如：`FILE *fp; fp=fopen(“file1.txt” , “r”);`

❑ 如果成功打开，返回一个指向被打开文件的文件信息区的起始地址；如果打开失败，返回一个NULL指针

§ 14.3 文件的打开与关闭

文件打开方式	含义
"r" (只读)	为输入打开一个文本文件
"w" (只写)	为输出打开一个文本文件
"a" (追加)	为追加打开一个文本文件
"rb" (只读)	为输入打开一个二进制文件
"wb" (只写)	为输出打开一个二进制文件
"ab" (追加)	为追加打开一个二进制文件
"r+" (读写)	读 / 写打开一个文本文件 (不创建)
"w+" (读写)	读 / 写创建一个文本文件 (清空写、创建)
"a+" (读写)	读 / 写打开一个文本文件 (追加写、创建)
"rb+" (读写)	为读 / 写打开一个二进制文件
"wb+" (读写)	为读 / 写创建一个二进制文件
"ab+" (读写)	为读 / 写打开一个二进制文件

§ 14.3 文件的打开与关闭

□ 关闭文件的作用：

- (1) 使文件指针fp与文件脱离
- (2) 刷新文件输入 / 输出缓冲区

□ 关闭方法：

`fclose(fp);`

§ 14.4 文件的读写

□ fputc() / fgetc()

- ✓ `int fputc(int c, FILE *fp)`; 把字符c写入文件fp, 成功时返回字符c的ASCII码; 失败时返回EOF (在stdio.h中, 符号常量EOF的值等于-1)
- ✓ `int fgetc(FILE *fp)`; 从文件fp中读1个字符, 返回读得的字符。对文本文件, 遇文件尾时返回EOF。

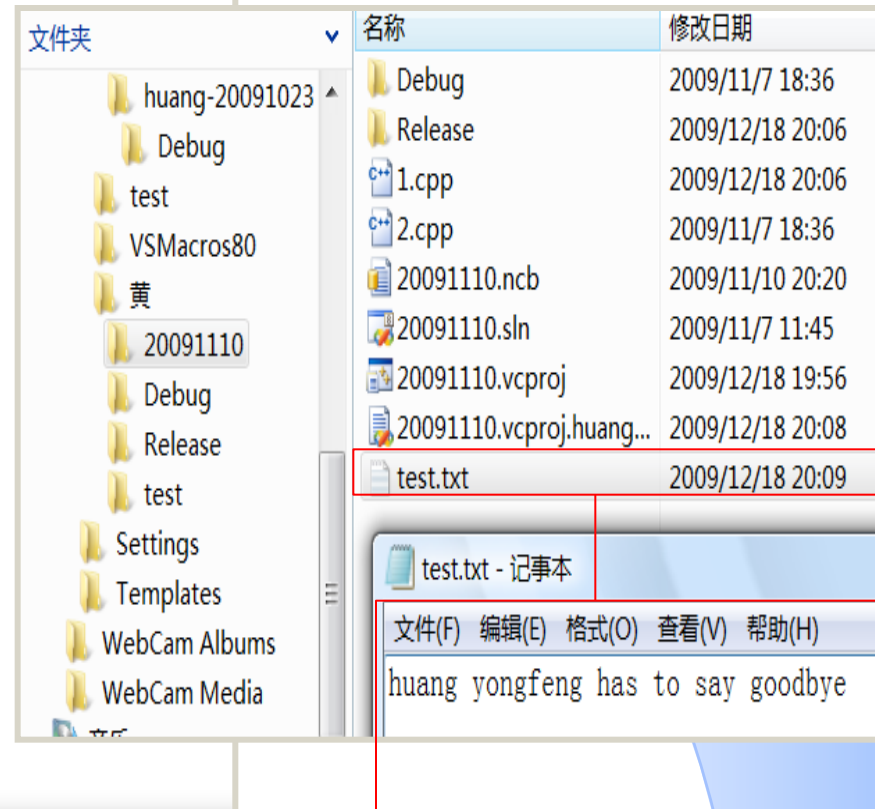
- 对于二进制文件, 可以用`feof(fp)` 判别是否遇文件尾。
`feof(fp)=1`说明遇文件尾。

从文本文件test.txt中顺序读入内容，并在屏幕上显示出来

```

1  #include "stdio.h"
2  int main()
3  {FILE *fp;
4   char ch;
5   fp=fopen("test.txt", "r");
6   if(fp==NULL)
7   {printf("can not open test\n");
8   return -1;}
9   ch=fgetc(fp);
10  while(ch!= EOF)
11  {putchar(ch);
12   ch = fgetc(fp);}
13  fclose(fp);
14  return 0;}
15

```



名称	修改日期
Debug	2009/11/7 18:36
Release	2009/12/18 20:06
1.cpp	2009/12/18 20:06
2.cpp	2009/11/7 18:36
20091110.ncb	2009/11/10 20:20
20091110.sln	2009/11/7 11:45
20091110.vcproj	2009/12/18 19:56
20091110.vcproj.huang...	2009/12/18 20:08
test.txt	2009/12/18 20:09

test.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

huang yongfeng has to say goodbye

C:\Windows\system32\cmd.exe

huang yongfeng has to say goodbye 请按

从键盘输入字符，逐个把它们写入磁盘文件，直到输入#止

```

1  #include "stdio.h"
2  int main()
3  {FILE *fp;
4    char ch;
5    char filename[10];
6    printf("Input filename\n");
7    scanf("%s\n", filename);
8    if ((fp = fopen(filename, "w"))==NULL)
9    {printf("can not open file %s\n", filename);
10     return 0;}
11    ch = getchar();
12    while(ch != '#')
13    {fputc(ch, fp);
14     putchar(ch);
15     ch = getchar();}
16    fclose(fp);
17

```

Input filename

huang

huang say goodbye#

huang say goodbye请按任意键

微软拼音输入法 2003 半 :

huang - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

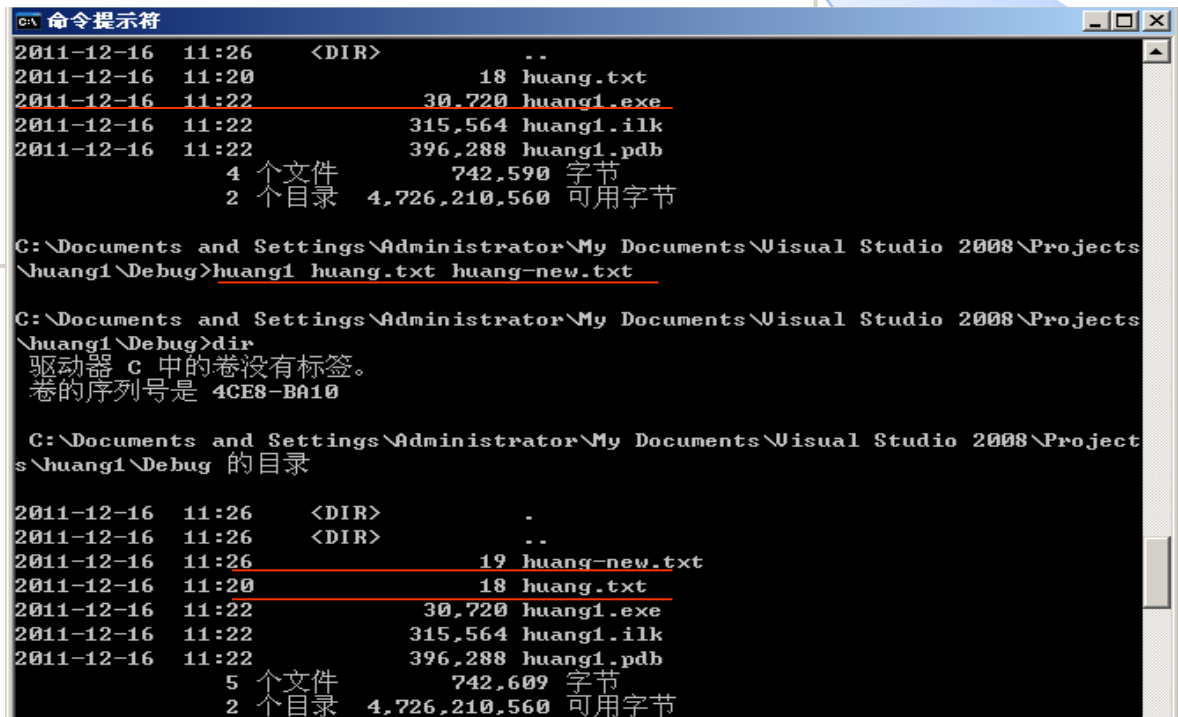
huang say goodbye

将一个磁盘文件的内容复制到另一个磁盘文件

```

1  #include "stdio.h"
2  int main(int argc, char *argv[ ])
3  { FILE *in, *out;
4    char ch;
5    if (argc<3)
6    {printf("Enter format is : huang1 source file name destination file name");
7    return 0;}
8    if ((in = fopen(argv[1], "rb"))==NULL)
9    {printf("can not open infile %s\n", argv[1]);
10   return(0);}
11   if ((out = fopen(argv[2], "wb"))==NULL)
12   {printf("can not open outfile %s\n", argv[2]);
13   return(0); }
14   while(!feof(in))
15   fputc(fgetc(in), out);
16   fclose(in); fclose(out);
17   return 0;
18 }

```



```

c:\ 命令提示符
2011-12-16 11:26 <DIR>      ..
2011-12-16 11:20          18 huang.txt
2011-12-16 11:22          30,720 huang1.exe
2011-12-16 11:22        315,564 huang1.ilc
2011-12-16 11:22        396,288 huang1.pdb
                4 个文件          742,590 字节
                2 个目录    4,726,210,560 可用字节

C:\Documents and Settings\Administrator\My Documents\Visual Studio 2008\Projects\
huang1\Debug>huang1 huang.txt huang-new.txt

C:\Documents and Settings\Administrator\My Documents\Visual Studio 2008\Projects\
huang1\Debug>dir
驱动器 c 中的卷没有标签。
卷的序列号是 4CE8-BA10

C:\Documents and Settings\Administrator\My Documents\Visual Studio 2008\Project
s\huang1\Debug 的目录

2011-12-16 11:26 <DIR>      .
2011-12-16 11:26 <DIR>      ..
2011-12-16 11:26          19 huang-new.txt
2011-12-16 11:20          18 huang.txt
2011-12-16 11:22          30,720 huang1.exe
2011-12-16 11:22        315,564 huang1.ilc
2011-12-16 11:22        396,288 huang1.pdb
                5 个文件          742,609 字节
                2 个目录    4,726,210,560 可用字节

```

§ 14.4 文件的读写

❑ fread() 读文件（块读）

- ✓ `size_t fread(void *buffer, size_t size, size_t count, FILE *fp);`
- ✓ 从文件fp中读入count次、每次读size字节、读入的信息存在buffer缓冲区。函数返回值等于实际读入的次数（可能少于count）。失败返回0

❑ fwrite() 写文件（块写）

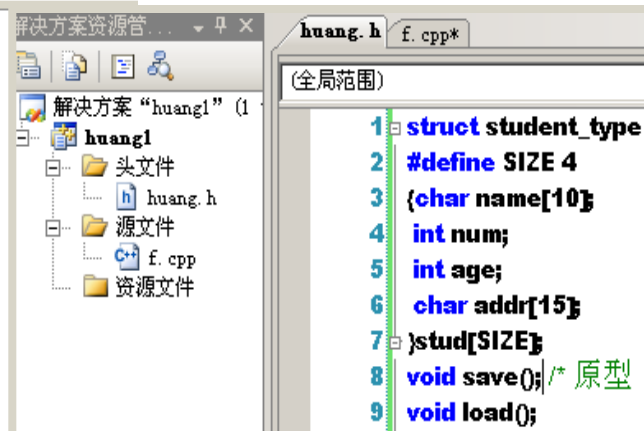
- ✓ `size_t fwrite(void *buffer, size_t size, size_t count, FILE *fp);`
- ✓ 将buffer中的信息、写入count次、每次写size字节至文件fp中。函数返回值等于实际写入的次数（可能少于count）。失败返回0

从键盘上输入一批学生数据，然后存储到磁盘上

```

1 #include "huang.h"
2 #include "stdio.h"
3 #include "stdlib.h"
4 int main()
5 {int i;
6 for(i=0; i<SIZE; i++)
7 scanf("%s%d%d%s", stud[i].name, &stud[i].num, &stud[i].age, stud[i].addr);
8 save(); /* 存盘 */
9 load(); /* 从盘读出 */
10 for(i=0; i<SIZE; i++)
11 printf("%-10s%-4d%-4d%-15s\n", stud[i].name, stud[i].num, stud[i].age, stud[i].addr);}
12 void save()
13 {FILE *fp;
14 int i;
15 if ((fp=fopen("stu_list", "wb"))==NULL)
16 {printf("can not open file\n");
17 exit(0);}
18 for(i=0; i<SIZE; i++)
19 if (fwrite(&stud[i], sizeof(struct student_type), 1, fp) == 0) printf("File write error\n");}
20 void load()
21 { FILE *fp; int i;
22 if ((fp=fopen("stu_list", "rb"))==NULL)
23 { printf("can not open file\n");
24 exit(0);}
25 for(i=0; i>SIZE; i++)
26 if (fread(&stud[i], sizeof(struct student_type), 1, fp) ==0)
27 { if (feof(fp))
28 exit(0);
29 printf("File read error\n");
30 }}

```



```

C:\WINDOWS\system32\cmd.exe
huang 1201 12 beijing
wang 1203 12 haidian
li 1204 13 qinghua
wangxia 1205 15 qinghua
huang 1201 12beijing
wang 1203 12haidian
li 1204 13qinghua
wangxia 1205 15qinghua
请按任意键继续. . .

```

§ 14.4 文件的读写

□ 其它读写磁盘函数

✓ `fprintf()` / `fscanf()` 函数（格式读写）

`//fprintf(文件指针, 格式控制, 变量列表);`

`//fscanf(文件指针, 格式控制, 变量地址列表);`

`//除增加“文件指针”外, 与printf() / scanf()用法相同`

✓ `int putw(int w, FILE *fp) ;` // 将整数w写入文件fp中

✓ `int getw(FILE *fp) ;` //从文件fp中读一个整数

✓ `char * fgets(char *str, int n, FILE *fp);` // 从文件fp读n-1个字节到str, str最后一个字节加 ‘\0’

✓ `int fputs(const char *str, FILE *fp)` //把str写入fp

§ 14.5 文件的定位

- 文件指针FILE *fp中，包含一个读写位置指针char *_nextc，它指向下一次文件读写的位置

```
typedef struct
{ int _fd;          /* 文件号 */
  int _cleft;       /* 缓冲区中剩下的字节数 */
  int _mode;        /* 文件操作模式 */
  char *_nextc;     /* 下一个字节的位置 */
  char *_buff;      /* 文件缓冲区位置 */
} FILE;
```

- 每当进行一次读写后，该指针自动指向下一次读写的位置；
- 当文件刚打开或创建时，该指针指向文件的开始位置；
- 可用函数ftell()获得当前的位置指针，也可用rewind() / fseek()函数改变位置指针，使其指向需要读写的位置；
- 原型: long ftell (FILE *fp); //文件fp的当前位置指针。若返回-1则失败。

§ 14.5 文件的定位

□ 原型: `void rewind(FILE *fp)`

作用: 使文件`fp`的位置指针指向文件开始。

□ 把一个文件的内容显示在屏幕, 同时复制到另一个文件

```
#include "stdio.h"
```

```
void main()
```

```
{FILE *fp1, *fp2;
```

```
fp1 = fopen("file1.c", "rb"); /* 源文件 */
```

```
fp2 = fopen("file2.c", "wb"); /* 复制到file2.c */
```

```
while(!feof(fp1)) putchar(fgetc(fp1)); /* 显示到屏幕上 */
```

```
rewind(fp1); /* fp回到开始位置 */
```

```
while(!feof(fp1)) fputc(fgetc(fp1), fp2);
```

```
fclose(fp1);
```

```
fclose(fp2);}
```

§ 14.5 文件的定位

- 原型： `int fseek (FILE *fp, long offset, int origin);`
 功能：把文件fp的位置指针从起始点origin开始，移动offset字节。成功返回当前位置，失败返回非-1。

起始点origin	符号常量	值
文件开始位置	SEEK_SET	0
当前位置	SEEK_CUR	1
文件尾	SEEK_END	2

在磁盘文件stud_dat上，存有10个学生(0~9)的数据，读出1、3、5、7、9号学生数据，并在屏幕上显示出来

```

1  #include "stdio.h"
2  struct student_type
3  {char name[10];
4    int num;
5    int age;
6    char sex;
7  }stu[10];
8  int main()
9  {int i;
10   FILE *fp;
11   if((fp=fopen("stud_dat", /*"rb"*/"w+")) == NULL)
12   {printf("can not open file\n");
13    return 0;}
14   for(i=0; i<10; i+=2)
15   {fseek(fp, i*sizeof(struct student_type), SEEK_SET);
16    fread(&stu[i], sizeof(struct student_type), 1, fp);
17    printf("%s %d %d %c\n", stu[i].name, stu[i].num, stu[i].age, stu[i].sex);
18  }
19  fclose(fp);

```

0 0 a
0 0 a
0 0 a
0 0 a
0 0 a
请按任意键继续.

微软拼音输入法 2003 半 :

20091110.vcproj.huang...	2009/12/18 20:40	Visual St
stud_dat	2009/12/18 20:40	文件
test.txt	2009/12/18 20:09	文本文档
test1	2009/12/18 20:25	文件

```


1 #include<stdio.h>
2 #include<stdlib.h>
3 void main()
4 {FILE *fp;
5  char c,ch[20];
6  int i;
7  if((fp=fopen("jean1.txt","r+"))==NULL)
8  {
9      printf("cannot open!\n");
10     exit(0);
11 }
12 printf("input string:");
13 c=getchar();
14 while(c!='*')
15 {
16     fputc(c,fp);
17     c=getchar();
18 }
19 rewind(fp);
20 //fclose(fp);
21 //fp=fopen("jean1.txt","r+");
22 for(i=0;i<20;i++)
23 fscanf(fp,"%c",&ch[i]);
24 for(i=0;i<20;i++)
25 printf("%c",ch[i]);
26 fclose(fp);
27 }
28

```

自动窗口		
名称	值	类型
ch	"黄永峰老师黄永峰老师"	char[]
[0]	-69 "	char
[1]	-58 "	char
[2]	-45 "	char
[3]	-64 "	char
[4]	-73 "	char
[5]	-27 "	char
[6]	-64 "	char
[7]	-49 "	char
[8]	-54 "	char
[9]	-90 "	char
[10]	-69 "	char
[11]	-58 "	char
[12]	-45 "	char
[13]	-64 "	char
[14]	-73 "	char
[15]	-27 "	char
[16]	-64 "	char
[17]	-49 "	char

自动窗口 局部变量 线程 模块 监视 1

请分析该程序运行的结果？

 `int fflush(FILE *fp) //清空文件I/O缓冲区流，出错则-1L`

```

3  #include <stdio.h>
4  #include <stdlib.h>
5  void main( )
6  { FILE *fp;
7    char c[21];
8    int i;
9    if((fp=fopen("r1.txt","w+"))==NULL)
10   { printf("cannot open this file!");
11     exit(0);
12   }
13   for(i=0; i<2; i++)
14   fputs("huang yong", fp);
15   fseek(fp, -15L, SEEK_CUR );
16   fputs(" YONG", fp);
17   fflush( fp ); /* 关键是需要这个函数清空缓冲区 */
18   // rewind(fp);
19   fgetc(c,21, fp);
20   c[20] = '\0';
21   puts(c);
22   fclose(fp);
23 }
24

```

请分析该程序运行的结果？

C:\WINDOWS\system32\cmd.exe

huang yong
请按任意键继续. . .

C:\WINDOWS\system32\cmd.exe

yonghuang yong 屯屯?
请按任意键继续. . .

C:\WINDOWS\system32\cmd.exe

huang YONGhuang yong
请按任意键继续. . .

计算机的I/O操作小结



scanf()
cin
getchar()
gets()



printf()
cout
putchar()
puts()



fprintf()
fputw()
fputc()
fputs()
fwrite()



文件

fscanf()
fgetc()
fgetw()
fgets()
fwrite()

课后必做题3道—在笔试之前提交

1. 在上一讲的课后练习第1题的基础上，完成文件读写操作，即上讲ppt的课后习题第1题中的（4）、（5）问（带*）

2. 编写一个C程序，从键盘输入一个字符串（输入的字符串以“#”作为结束），将其中的小写字母全部转换成大写字母，并写入到文件 upper.txt 中。然后再从该文件中的内容读出并显示输出。

3. 编写一个C程序，主函数从命令行得到一个文件名，然后调用函数 `fgets()` 从文件中读入一字符串存放到字符数组 `str` 中（字符个数最多为 80 个）。在主函数中输出字符串与该字符串的长度。`fgets` 函数的格式为：`char *fgets(char *string, int n, FILE *stream);`

选做题1道

1. 假设有学生情况如右边的表格所示，要求按照表格中的数据，编程完成如右边（1）—（8）的操作。

表 X.1 学生情况登记表

学号(num)	姓名(name)	性别(sex)	年龄(age)	成绩(grade)
101	Zhang	M	19	95.6
102	Wang	F	18	92.4
103	Zhao	M	19	85.7
104	Li	M	20	96.3
105	Gao	M	19	90.2
106	Lin	M	18	91.5
107	Ma	F	17	98.7
108	Zhen	M	21	90.1
109	Xu	M	19	89.5
110	Mao	F	18	94.5

编写一个C程序，依次实现以下操作：

(1) 定义一个结构体类型

```
struct student
{
    char num[7];
    char name[8];
    char sex[3];
    char age[5];
    char grade[9];
};
```

(2) 为表 X.1 定义一个结构体类型(struct student)数组，并进行初始化。

(3) 打开一个可读写的新文件 stu.dat。

(4) 用函数 fwrite() 将结构体数组内容写入文件 stu.dat 中。

(5) 关闭文件 stu.dat。

(6) 打开可读写文件 stu.dat。

(7) 从文件 stu.dat 中读出各学生情况并输出。输出格式如表 X.1 所示，但不要表格框线。

(8) 关闭文件 stu.dat。

重要通知

- 15周上机考试，分5个单元时间考试，具体时间同期中机考；3道题，20分；线上考试。具体规则见下页。
- 期末笔试考试安排：时间12月30；具体考试规则见另外通知。

期末机考安排

- 本学期期末机考全部采取线上考试的形式，共分为5个时间段：
周二（12月20日）中午：11:30开始准备，12:00开始，14:30结束。
周二（12月20日）下午：15:00开始准备，15:30开始，18:00结束。
周二（12月20日）晚上：18:30开始准备，19:00开始，21:30结束。
周三（12月21日）晚上：18:30开始准备，19:00开始，21:30结束。
周五（12月23日）晚上：18:30开始准备，19:00开始，21:30结束。
- 目前按照二级选课的时间段进行分组，具体分配情况请参考课程文件。
如有课程冲突或其他有必要调换时间的特殊情况，请向ybch14@163.com发送邮件，注明姓名、学号、原时间段、调整后时间段、需要调整的必要性理由。【时间调换截止时间本周三（12月14日）23:59（以收到邮件时间为准）】。请各位同学相互转告！之后将对每个场次建立微信群发布后续通知。
- 每位同学只允许进行一次上机考试。如有同学进行多次考试，核实后，将按照所有考试成绩的最低分进行登记。
- 考试范围：截止到文件之前（不包括文件），其中数组、指针、字符串为重点，函数递归也可能会涉及。

期末机考安排

评分规则：

在2小时正式考试时间内完成并一次性通过验收者得满分20分。

在2小时到2小时15分钟内完成者，扣2分。

在2小时15分钟到2小时30分钟内完成者，扣4分。

2小时30分钟未完成者酌情评分。

此外，不论正式考试时间或延长时间，每次验收失败额外扣2分。

考试纪律：

考试时，每个人自己独立完成考题的编程和调试。禁止对外求助，线上机考具体事宜会后续在微信群组说明。可以使用教材等参考书籍。请各位同学严格遵守清华大学教育教学纪律，一旦发现作弊等不端行为，将按照清华大学校规严肃处理。