**学生成绩管理系统**

**姓 名： 李家玺**

**学 号： 2020404164**

**专 业： 计算机科学与技术**

**年 级： 20级**

2022年 6 月 9 日

目录

[**1．问题描述 3**](#_Toc26264)

[**2．需求分析 4**](#_Toc25217)

[2.1软件的基本功能 4](#_Toc9561)

[2.2输入/输出形式 4](#_Toc28042)

[2.3测试数据要求 4](#_Toc28110)

[**3．概要设计 4**](#_Toc3660)

[3.1所需的ADT及作用 4](#_Toc864)

[3.2主程序流程及模块调用关系 5](#_Toc8323)

[**4．详细设计 6**](#_Toc11652)

[4.1实现概要设计的数据类型 6](#_Toc4114)

[4.2主程序和其它模块的算法 7](#_Toc32084)

[4.3主程序和其它模块的编码实现 12](#_Toc32639)

[4.4函数调用关系图 16](#_Toc16668)

[**5．编码与调试分析 16**](#_Toc29212)

[**6.使用说明 21**](#_Toc21256)

[**7.测试结果 21**](#_Toc5456)

[**8.课程设计心得体会 26**](#_Toc29465)

[**附录：（其他各模块编码实现） 27**](#_Toc31845)

**1．问题描述**

学生成绩管理

现有学生成绩信息文件1（1.txt），内容如下

姓名    学号   语文  数学   英语

张明明  01    67   78   82

李成友  02    78   91   88

张辉灿  03    68   82   56

王露    04    56   45   77

陈东明  05    67   38   47

….    ..    ..   ..   …

学生成绩信息文件2（2.txt）,内容如下:

姓名    学号   语文  数学   英语

陈果    31    57   68   82

李华明  32    88   90   68

张明东  33    48   42   56

李明国  34    50   45   87

陈道亮  35    47   58   77

….    ..    ..   ..   …

试编写一管理系统,要求如下:

1)实现对两个文件数据进行合并,生成新文件3.txt

2)抽取出三科成绩中有补考的学生并保存在一个新文件4.txt

3)对合并后的文件3.txt中的数据按总分降序排序(至少采用两种排序方法实现)

4)输入一个学生姓名后,能查找到此学生的信息并输出结果(至少采用两种查找方法实现)

5)要求使用结构体,链或数组等实现上述要求

**2．需求分析**

2.1软件的基本功能

随着在校大学生人数的不断增加，教务系统的数据量也不断的上涨。学校工作繁杂，资料众多，虽然各类管理信息系统已进入高校，但还未普及，而对于学生成绩管理来说，目前还没有一套完整的，一的系统因此开发一套适合大众的，兼容性好的系统是很有必要。通过开发学生成绩管理系统，使用计算机对学生成绩信息进行管理，具有手工管理所无法比拟的检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成本低等优点，可以提高校务人员的工作效率和学生的查询效率。此设计能够存储每个学生的成绩信息，并完成合并txt文件中的学生成绩信息，并对学生信息进行排序、抽取不合格学生和查找学生信息等功能。为了方便管理学生成绩，进行对成绩的统计和计算排序等功能设计此系统。

2.2输入/输出形式

数据以文件形式按一定格式要求输入和输出。

2.3测试数据要求

学生成绩信息分别存储在1.txt文件和2.txt文件中，测试数据可以在文件中直接按一定格式输入，然后载入程序；也可以在程序中按功能提示添加。

**3．概要设计**

3.1所需的ADT及作用

学生信息用结构体实现，这样有利于封装每一个实体对象，并且方便数据操作。

用户信息采用结构体数组存储，因为在程序中不涉及用户信息的修改，用该数据结构便于遍历用户信息，提高程序运行效率。

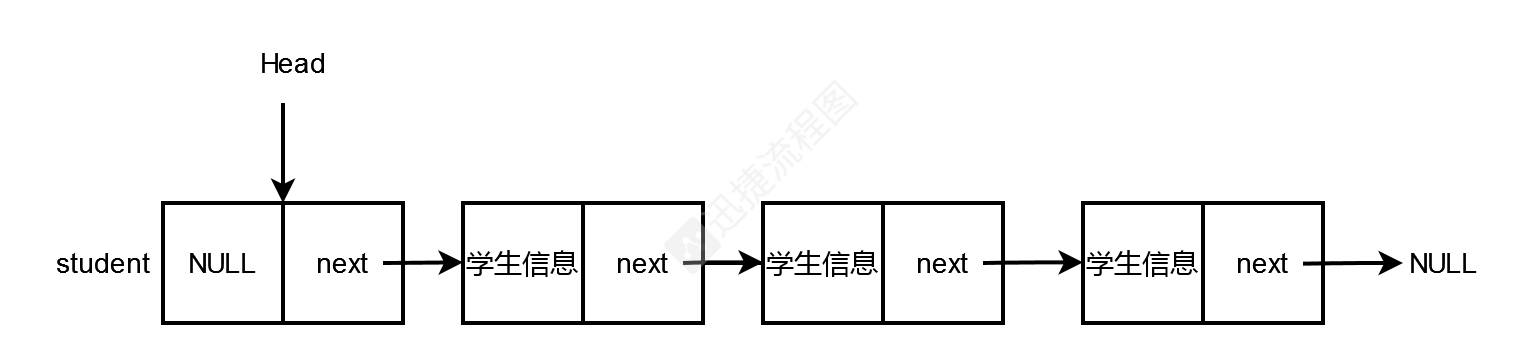


图1 学生结构体

学生信息因不涉及到数据添加和删除，用数组存储，方便查找。

3.2主程序流程及模块调用关系

主流程图：

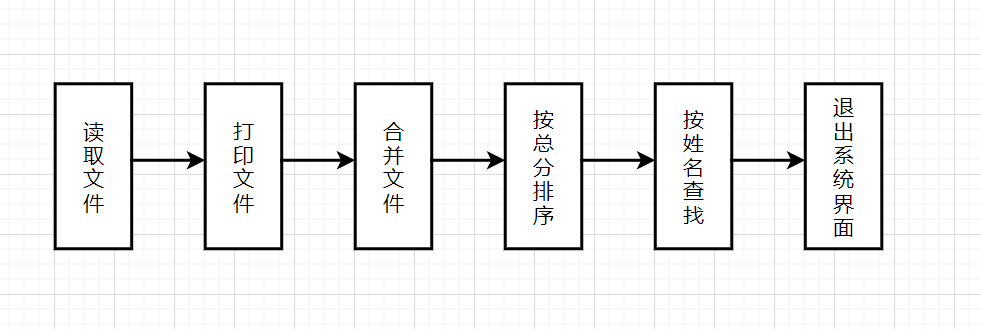


图2 主流程图

模块调用关系：

系统分为6个模块：主模块，decsSort排序，fileMerge合并，fileStore补考学生存储search查找，student学生。

表1 各模块函数及其功能

|  |  |
| --- | --- |
| 功能 | 函数 |
| 主函数 | int main() |
| 合并文件 | void link() |
| 排序 | 选择排序：void selectSort()；冒泡排序：void bubbleSort() |
| 查询 | void search1(char \*name)；void search2(char \*name){ |
| 筛选补考学生 | void bu\_kao() |

主模块：调用其他各个模块，组成完整的系统，显示部分提示信息。

decsSort排序：该模块包含根据学生总分对学生进行重新排序并且重新存入3.txt文件中的功能。

fileMerge合并：该模块用于实现1.txt文件和2.txt文件的合并，并存储到新建的3.txt 文件中。

fileStore补考学生存储：该模块用于找出学生信息中需要补考的学生信息，并将查询 出的信息存储到一个新的4.txt文件中。

search查找：该模块用于依据用户所输入的学生姓名，查找到此学生的信息并输出。

student学生：该模块用于1.txt文件和2.txt文件中的学生信息数据进行录入，并打印出来。

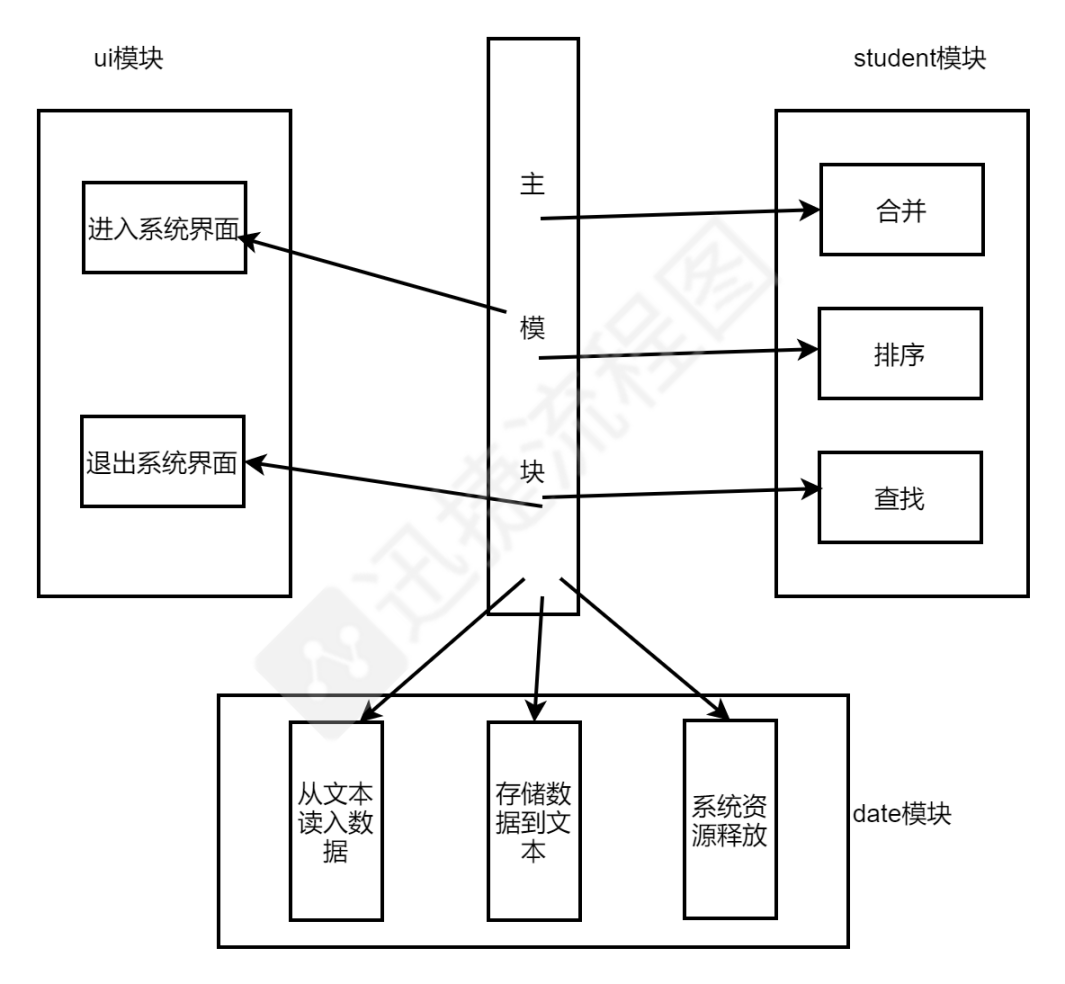


图3 函数调用关系图

**4．详细设计**

4.1实现概要设计的数据类型

//学生信息

typedef struct Student{

char sname[20];

int sno;

int chinese;

int math;

int english;

struct Student \*next;

}Student,\*gradelist;

4.2主程序和其它模块的算法

由于学生信息成绩管理系统的主程序主要用于调用其他各个模块，组成完整的系统，显示部分提示信息，不测及到算法。

//按学生成绩总分排序

//方法1：选择排序

void selectSort(){

char \*ad3="3.txt";//一、对文件进行处理 找出文件3的地址

gradelist f3=firead(ad3);//将文件3的内容存入结构f3中

Student \*p=f3;

if(remove("3.txt")){//将文件3内容清空//先将文件3删除 使用remove函数

printf("删除失败");

exit(0);

}

FILE \*fp;

if((fp=fopen("3.txt","w+"))==NULL){//再新建文件3

printf("新建失败");

exit(0);

}

fprintf(fp,"%s",bt);//文件3标题

int stucnt=0;//计算学生个数

while(p->next!=NULL){

stucnt++;

p=p->next;

}

typedef struct Stu{//二、建立局部结构体方便计算,用数组需要大量数组存储数据//存储数据

char sname[20];

int sno;

int chinese;

int math;

int english;

int totalscore; //新加入总分，方便计算

}gradeFlag;

typedef struct grade{//排序结构数组，方便输出

gradeFlag g[500];

}gradeTable;

gradeTable T; //设置T整体结构数组

p=f3;//将数据放入T结构数组中

int i;

for(i=1;i<=stucnt;i++,p=p->next){

strcpy(T.g[i].sname,p->sname);

T.g[i].sno=p->sno;

T.g[i].chinese=p->chinese;

T.g[i].math=p->math;

T.g[i].english=p->english;

T.g[i].totalscore=p->chinese+p->math+p->english;

}

int maxx=0; //三、开始排序(选择排序)//用来记住最大数字的位置下表标

int t,j,w; //循环计数

int temp=stucnt+1;

for(i=0;i<stucnt;i++){

maxx=i;

for(j=i+1;j<=stucnt;j++){

if(T.g[j].totalscore>T.g[maxx].totalscore){

maxx=j;

}

}

if(maxx!=i){

T.g[temp]=T.g[i];

T.g[i]=T.g[maxx];

T.g[maxx]=T.g[temp];

}

}

//数据写入文件

for(w=0;w<stucnt+1;w++){

fprintf(fp,"%5s %2d %d %d %d\n",T.g[w].sname,T.g[w].sno,

T.g[w].chinese,T.g[w].math,T.g[w].english);

}

if(fclose(fp)){

printf("不能关闭文件!\n");

exit(0);

}

}

//方法2：冒泡排序

void bubbleSort(){

//一、对文件进行处理 找出文件3的地址

char \*ad3="3.txt";

//将文件3的内容存入结构f3中

gradelist f3=firead(ad3);

Student \*p=f3;

//将文件3内容清空

//先将文件3删除 使用remove函数

if(remove("3.txt")){

printf("删除失败");

exit(0);

}

FILE \*fp;

//再新建文件3

if((fp=fopen("3.txt","w+"))==NULL){

printf("新建失败");

exit(0);

}

//文件3标题

fprintf(fp,"%s",bt);

//计算学生个数

int stucnt=0;

while(p->next!=NULL){

stucnt++;

p=p->next;

}

//二、建立局部结构体方便计算,用数组需要大量数组存储数据

//存储数据

typedef struct Stu{

char sname[20];

int sno;

int chinese;

int math;

int english;

int totalscore; //新加入总分，方便计算

}gradeFlag;

//排序结构数组，方便输出

typedef struct grade{

gradeFlag g[500];

}gradeTable;

gradeTable T; //设置T整体结构数组

p=f3;

//将数据放入T结构数组中

int i;

for(i=1;i<=stucnt;i++){

//出现数据类型匹配错误

strcpy(T.g[i].sname,p->sname);

T.g[i].sno=p->sno;

T.g[i].chinese=p->chinese;

T.g[i].math=p->math;

T.g[i].english=p->english;

T.g[i].totalscore=p->chinese+p->math+p->english;

p=p->next;

}

int flag,nflag, tmp=stucnt+1;

flag=stucnt;

do{

nflag = 0;

for(i = 0; i < stucnt - 1; i++){

if(T.g[i].totalscore > T.g[i+1].totalscore){

T.g[tmp] = T.g[i];

T.g[i]= T.g[i+1];

T.g[i+1] = T.g[tmp];

nflag = i + 1;

}

}

stucnt = nflag;

}while(nflag);

int w;

for(w=flag-1;w>=0;w--){

fprintf(fp,"%5s %2d %d %d %d\n",T.g[w].sname,T.g[w].sno,

T.g[w].chinese,T.g[w].math,T.g[w].english);

}

if(fclose(fp)){

printf("不能关闭文件!\n");

exit(0);

}

}

因为需要按照学生成绩的总分进行查找，所以需将之前3.txt文件中的内容进行删除， 然后在重新建立3.txt文件，然后通过建立数组并将学生成绩总分加入，以便后面进行 查找。判断换位条件：按照学生成绩总分大小进行降序排序。其他的实现思想分别和选 择排序、冒泡排序一样。

4.3主程序和其它模块的编码实现

主模块

#include <stdio.h>  
#include <stdlib.h>  
#include "decsSort.h"  
#include "fileMerge.h"  
#include "fileStore.h"  
#include "score.h"  
#include "search.h"  
#include "student.h"  
  
  
int main()  
{  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("                   学生成绩管理系统                    \n");  
    printf("                      2022/6/10                        \n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("                      请选择功能                       \n");  
    printf(" 1.显示已有信息\n 2.合并文件\n 3.查找需要补考的学生\n 4.成绩排序\n 5.查找学生\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    system("pause");  
      
    gradelist file1=firead("1.txt");  
    gradelist file2=firead("2.txt");  
      
    int choice;  
    while(choice!=-1){  
    scanf("%d",&choice);  
    switch(choice){  
    case 1:  
  
        printf("成绩信息一\n");  
        showFile(file1);  
        printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
        printf("成绩信息二\n");  
        showFile(file2);  
        printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
        break;  
  
    case 2:  
        //system("pause");  
        printf("实现对两个文件数据进行合并，生成新文件3 \n");  
    link();  
    printf("合并成功\n");  
  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    gradelist file3=firead("3.txt");  
    showFile(file3);  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    break;  
  
    case 3:  
        printf("抽取出三科成绩中有补考的学生并保存在一个新文件4 \n");  
    bu\_kao();  
    //system("pause");  
    printf("保存成功\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    gradelist file4=firead("4.txt");  
    showFile(file4);  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    break;  
  
    case 4:  
        printf("对合并后的文件3.txt中的数据按总分降序排序 \n");  
    printf("方法一 冒泡排序\n");  
    bubbleSort();  
    file3=firead("3.txt");  
    printf("已排序\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    showFile(file3);  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    printf("方法二 选择排序\n");  
    selectSort();  
    file3=firead("3.txt");  
    printf("已排序\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    showFile(file3);  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    break;  
  
    case 5:  
         printf("输入一个学生姓名，查找此学生信息并输出 \n");  
    char name1[20];  
    printf("请输入要查找的学生姓名\n");  
    scanf("%s",&name1);  
    search(name1);  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    printf("输入一个学生姓名，查找此学生信息并输出 \n");  
    char name2[20];  
    printf("请输入要查找的学生姓名\n");  
    scanf("%s",&name2);  
    search2(name2);  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    break;  
  
    default:  
        break;  
    }  
    }  
    printf("程序结束\n");  
  
}

主模块调用其他各个模块实现系统组装

其他模块编码实现见附录源码

4.4函数调用关系图

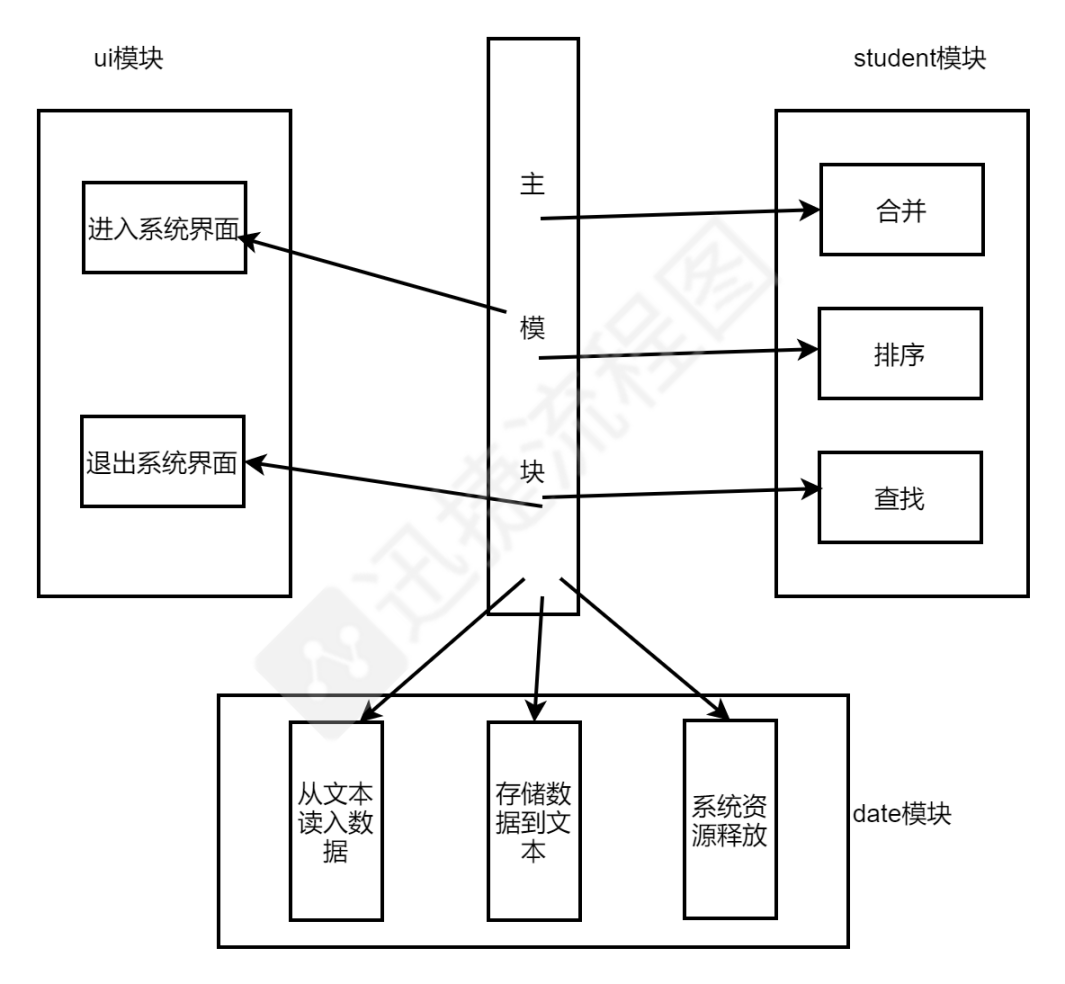


图4 函数调用关系图

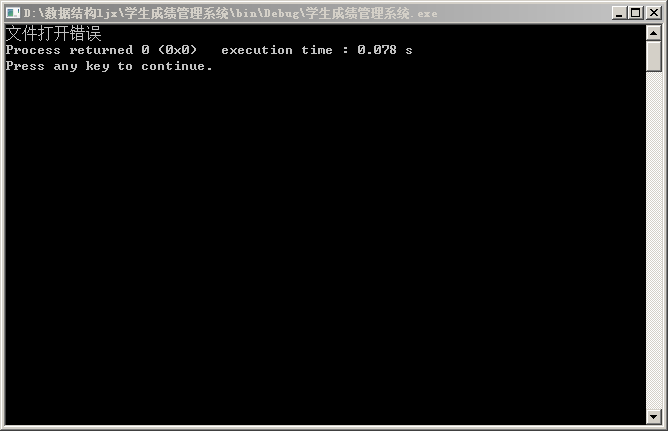
**5．编码与调试分析**

（1）运行各函数

出现fileread问题，函数名称为firead，无法识别。结构体Student，连接两表忘记存入第三表中，参数名称忘记同意（分工时）。

（2）运行主函数

a．出现文件无法打开



b.修改文件路径：将文件放入同一目录的文件夹，将路径改为文件名

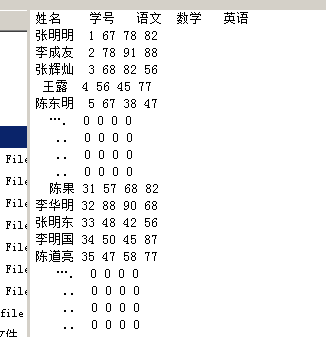


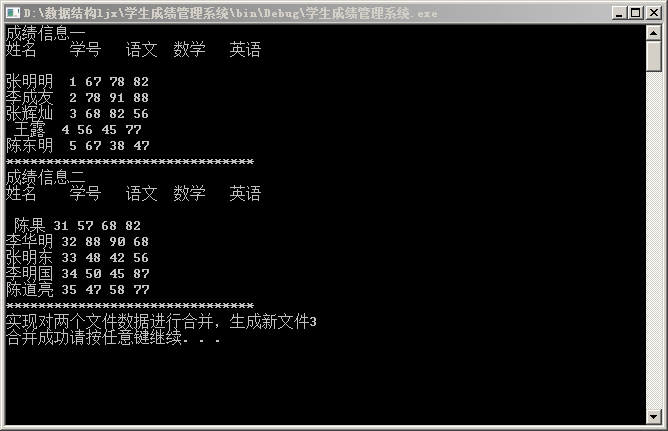
c.建立3.txt成功，但内容不太准确，并且成绩无法显示

修改fclose 关闭成功

显示成绩函数定义中未带上变量

文件1、2中删除数据后省略号，去除为0的数据



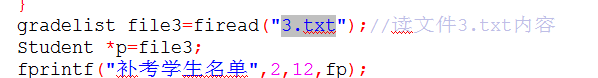


d.文件打开错误 修改成3.txt

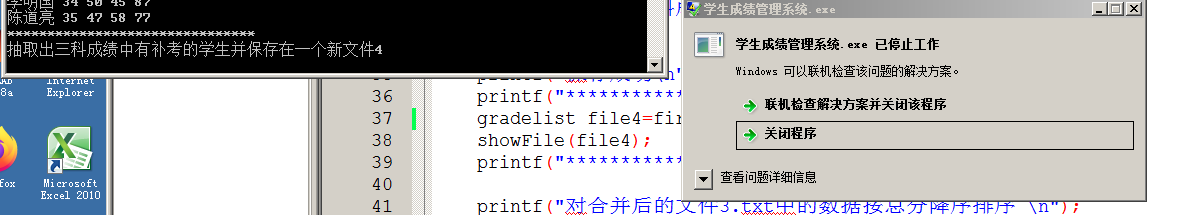


C:\Users\jd501\Documents\Tencent Files\240120511\FileRecv\MobileFile\Image\}MBSLBHEVN}BMGU1UW`L%SV.png

e.文件打开错误，忘记添加文件名



f.无法保存：fprint函数使用格式错误



g.无法运行exe文件，但是编程时未出错，几点原因：

①除以零

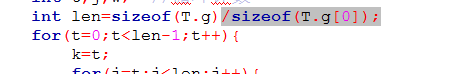
②数组越界：int a[3]; a[10000000]=10;

③指针越界：int \* p; p=(int \*)malloc(5 \* sizeof(int)); \*(p+1000000)=10;

④使用已经释放的空间：int \* p; p=(int \*)malloc(5 \* sizeof(int));free(p); \*p=10;

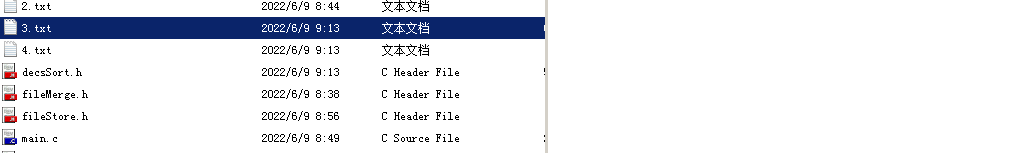
⑤数组开得太大，超出了栈的范围，造成栈溢出：int a[100000000]

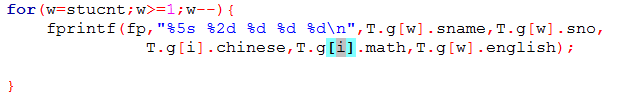
问题在于出栈 sizeof是求数组大小，sizeof/才是求数组长度

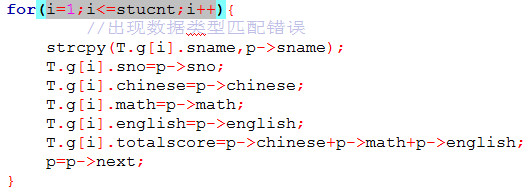


h.文件3中内容被删除，但未将排序后内容填入文件3

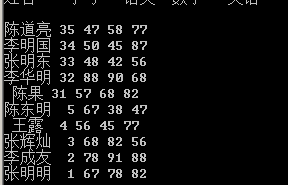
下标使用错误，未将内容与结构体一一对应

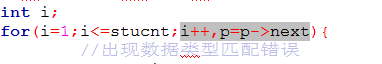






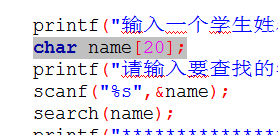
i.排序结果出错



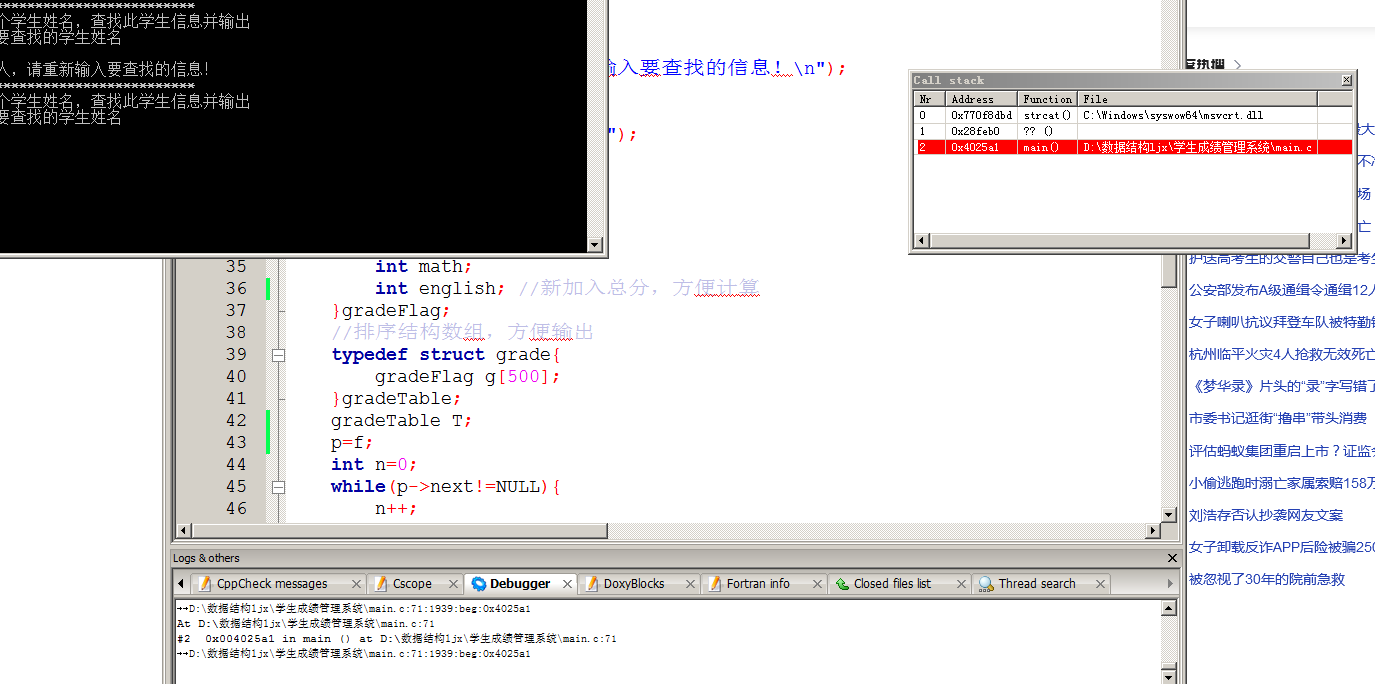


P为空值也应该设置为循环结束的条件，输出结果少一条，变为0000

j.无法使用查询函数，未将字符串设定空间



k.代码出现无法运行的问题，进行debug调试



**6.使用说明**

1.打开“学生成绩管理系统”

然后看见系统显示1.txt文件和2.txt文件中的学生信息

2.提示文件合并成功，生成新的3文件

按下任意键键，系统开始显示合并完成后3.txt文件中的内容

3.提示抽取出三科成绩中有补考的学生并保存在一个新文件4

补考学生名单请按任意键继续. . .

保存成功

按下任意键后显示抽取出的需要补考的学生信息。

4.提示对合并后的文件3.txt中的数据按总分降序排序

显示排好序后的学生成绩信息。

5.提示输入一个学生姓名，查找此学生信息并输出，请输入要查找的学生姓名

输入学生姓名如果不合理或不存在，系统将提示查无此人，请重新输入要查找的信息！

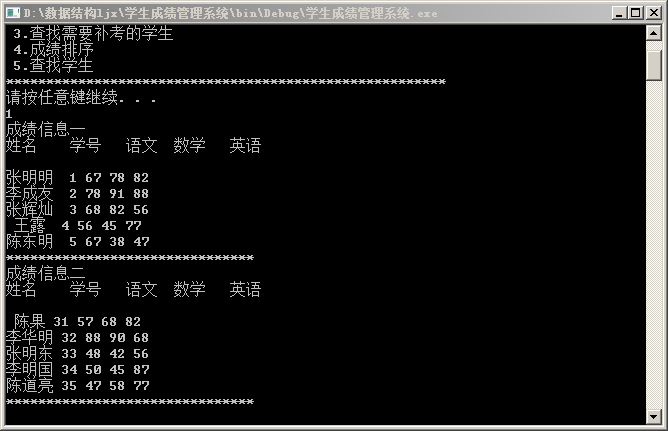
输入学生姓名存在且合理，系统将显示改名学生的信息。

**7.测试结果**

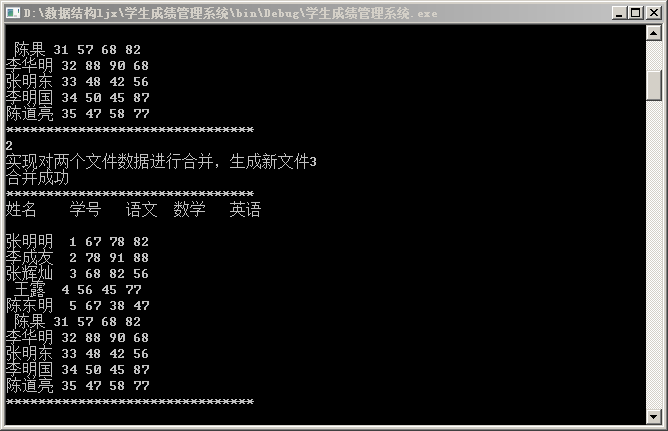
1）、打印菜单，提供功能选择



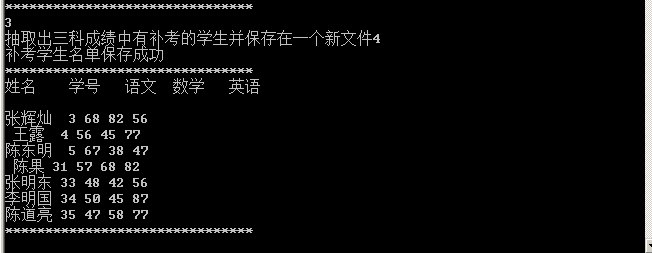
2）、选择功能1，显示1。txt文件和2.txt文件中的内容



3）、选择功能2，对1.txt文件和2.txt文件进行合并，并存入3.txt文件，显示3.txt 文件内容

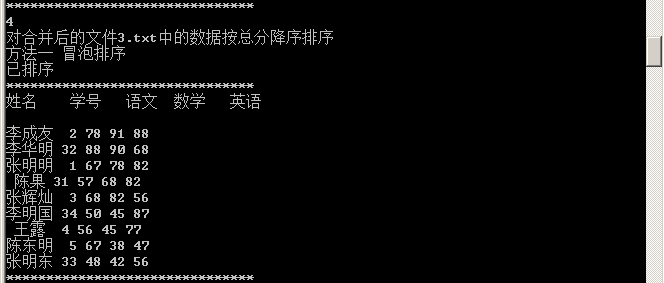


4）、选择功能3，抽取出学生中需要补考的学生信息，并存入4.txt文件中，显示其中 内容

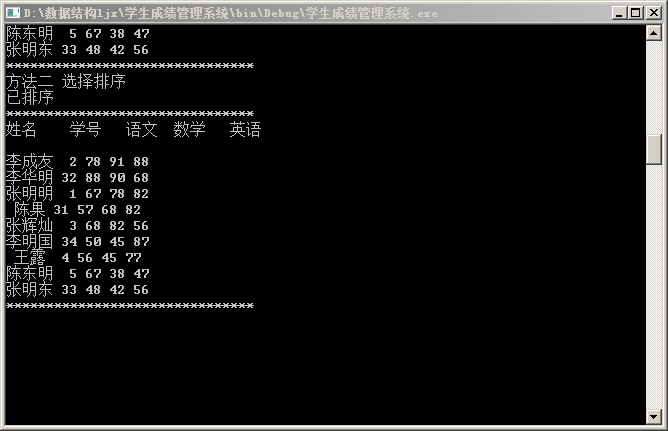


5）、选择功能5，对合并后的3.txt文件中的内容进行排序，并显示内容

方法一 冒泡排序

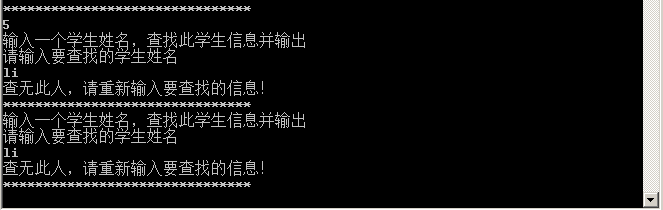


方法二 选择排序

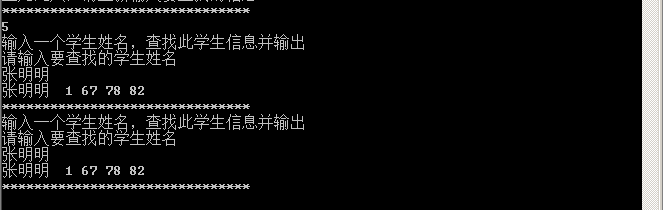
****

6）、根据姓名查询学生信息（两种方法）

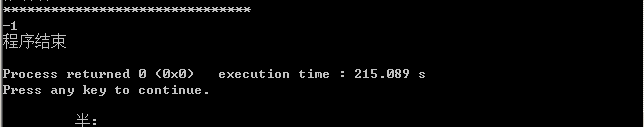
查询失败



查询成功



7）、输入-1，程序结束

****

**8.课程设计心得体会**

我们的这次课题是学生信息管理，针对具体的实际项目来进行需求分析，设计，测试等，使我们收获了很多。通过课程设计我们丰富了计算机操作经验,更加深了对数据结构的理解。

知道了编程开始要做的是先要了解整个思路框架，把整体框架先做出来，具体算法可以后面再加上，做完一部分就编译一次，把错误及时找出来，防止到最后因错误太多无从下手，避免了不必要的失误。

通过这次课程设计，我们深刻的认识到算法在程序中的重要性，一个完整的程序总是由若干个函数构成的，这些相应的函数体现了算法的基本思想，在学生信息管理程序中，有学生实体采用结构体数组，可以用每个节点存储每个个体的信息，操作简便。

求结构体长度本来用的是length，不过后来用的sizeof，但后来发现sizeof是求大小导致程序出现错误，后来通过查找资料发现应用sizeof（整个数组所占的内存字节）/sizeof（每个元素所占的字节）。

进行排序时，本想使用数组进行计算排序，后发现若要使用数组会产生很多数组进行存储，则更改使用结构体，只需两个结构体就可完成功能。排序过程中一开始未想到将文件3内容删除，导致结果一直在后来生成的文件3内容的基础上实现。排序过程中一直有循环变量取值的问题，逻辑不够清晰。可以看出，程序需要合理规范，不可有过多冗余的部分。  
 fopen函数在使用过程中发现许多问题，路径一开始无法打开文件，后在老师的提醒下放入同一目录，再将文件路径改为文件名，本以为是使用程序将内容输入文件，后发现审题不清。  
 查找学生程序部分第二种方法是基于排序结构体进行设计，但是缺点是并没有想到除了使用strcmp之外的方法，二分查找无法进行字符串对比，证明水平还不够，基础还不够扎实。

这次课程设计，让我们更加了解到数据结构的重要性，以及它对我们专业的发展发挥的重要作用。对我们而言，知识上的收获很重要，但精神上的收获更加重要。学到了课本上学习不到的内容，比如独立思考问题，出现差错的随机应变，这些都让我们收益匪浅。今后也能完成更加高质量的项目。

附录：（其他各模块编码实现）

decsSort排序

#ifndef DECSSORT\_H\_INCLUDED

#define DECSSORT\_H\_INCLUDED

#include <stdio.h>

#include <stdlib.h>

//#include "decsSort.h"

#include "fileMerge.h"

#include "fileStore.h"

#include "score.h"

#include "search.h"

#include "student.h"

//方法1：选择排序

void selectSort(){

//一、对文件进行处理 找出文件3的地址

char \*ad3="3.txt";

//将文件3的内容存入结构f3中

gradelist f3=firead(ad3);

Student \*p=f3;

//将文件3内容清空

//先将文件3删除 使用remove函数

if(remove("3.txt")){

printf("删除失败");

exit(0);

}

FILE \*fp;

//再新建文件3

if((fp=fopen("3.txt","w+"))==NULL){

printf("新建失败");

exit(0);

}

//文件3标题

fprintf(fp,"%s",bt);

//计算学生个数

int stucnt=0;

while(p->next!=NULL){

stucnt++;

p=p->next;

}

//二、建立局部结构体方便计算,用数组需要大量数组存储数据

//存储数据

typedef struct Stu{

char sname[20];

int sno;

int chinese;

int math;

int english;

int totalscore; //新加入总分，方便计算

}gradeFlag;

//排序结构数组，方便输出

typedef struct grade{

gradeFlag g[500];

}gradeTable;

gradeTable T; //设置T整体结构数组

p=f3;

//将数据放入T结构数组中

int i;

for(i=1;i<=stucnt;i++,p=p->next){

//出现数据类型匹配错误

strcpy(T.g[i].sname,p->sname);

T.g[i].sno=p->sno;

T.g[i].chinese=p->chinese;

T.g[i].math=p->math;

T.g[i].english=p->english;

T.g[i].totalscore=p->chinese+p->math+p->english;

}

//三、开始排序(选择排序)

int maxx=0; //用来记住最大数字的位置下表标

int t,j,w; //循环计数

//int len=sizeof(T.g)/sizeof(T.g[0]);

int temp=stucnt+1;

for(i=0;i<stucnt;i++){

maxx=i;

for(j=i+1;j<=stucnt;j++){

if(T.g[j].totalscore>T.g[maxx].totalscore){

maxx=j;

}

}

if(maxx!=i){

T.g[temp]=T.g[i];

T.g[i]=T.g[maxx];

T.g[maxx]=T.g[temp];

}

}

//数据写入文件

//fprintf(fp,"%s",bt);

for(w=0;w<stucnt+1;w++){

fprintf(fp,"%5s %2d %d %d %d\n",T.g[w].sname,T.g[w].sno,

T.g[w].chinese,T.g[w].math,T.g[w].english);

}

if(fclose(fp)){

printf("不能关闭文件!\n");

exit(0);

}

}

//方法2：冒泡排序

void bubbleSort(){

//一、对文件进行处理 找出文件3的地址

char \*ad3="3.txt";

//将文件3的内容存入结构f3中

gradelist f3=firead(ad3);

Student \*p=f3;

//将文件3内容清空

//先将文件3删除 使用remove函数

if(remove("3.txt")){

printf("删除失败");

exit(0);

}

FILE \*fp;

//再新建文件3

if((fp=fopen("3.txt","w+"))==NULL){

printf("新建失败");

exit(0);

}

//文件3标题

fprintf(fp,"%s",bt);

//计算学生个数

int stucnt=0;

while(p->next!=NULL){

stucnt++;

p=p->next;

}

//二、建立局部结构体方便计算,用数组需要大量数组存储数据

//存储数据

typedef struct Stu{

char sname[20];

int sno;

int chinese;

int math;

int english;

int totalscore; //新加入总分，方便计算

}gradeFlag;

//排序结构数组，方便输出

typedef struct grade{

gradeFlag g[500];

}gradeTable;

gradeTable T; //设置T整体结构数组

p=f3;

//将数据放入T结构数组中

int i;

for(i=1;i<=stucnt;i++){

//出现数据类型匹配错误

strcpy(T.g[i].sname,p->sname);

T.g[i].sno=p->sno;

T.g[i].chinese=p->chinese;

T.g[i].math=p->math;

T.g[i].english=p->english;

T.g[i].totalscore=p->chinese+p->math+p->english;

p=p->next;

}

int flag,nflag, tmp=stucnt+1;

flag=stucnt;

do{

nflag = 0;

for(i = 0; i < stucnt - 1; i++){

if(T.g[i].totalscore > T.g[i+1].totalscore){

T.g[tmp] = T.g[i];

T.g[i]= T.g[i+1];

T.g[i+1] = T.g[tmp];

nflag = i + 1;

}

}

stucnt = nflag;

}while(nflag);

int w;

for(w=flag-1;w>=0;w--){

fprintf(fp,"%5s %2d %d %d %d\n",T.g[w].sname,T.g[w].sno,

T.g[w].chinese,T.g[w].math,T.g[w].english);

}

if(fclose(fp)){

printf("不能关闭文件!\n");

exit(0);

}

}

#endif // DECSSORT\_H\_INCLUDED

fileMerge合并

#ifndef FILEMERGE\_H\_INCLUDED

#define FILEMERGE\_H\_INCLUDED

#include "decsSort.h"

//#include "fileMerge.h"

#include "fileStore.h"

#include "score.h"

#include "search.h"

#include "student.h"

void link()

{

char \*address1="1.txt",\*address2="2.txt",\*address3="3.txt";

gradelist f1=firead(address1), f2=firead(address2);

FILE \*fp;

if((fp=fopen("3.txt","w+"))==NULL)

{

printf("合并文件失败");

exit(0);

}

Student \*p1=f1,\*p2=f2;

fprintf(fp,"%s",bt);

while(p1->next!=NULL)

{

fprintf(fp,"%5s %2d %d %d %d\n",p1->sname,p1->sno,p1->chinese,p1->math,p1->english);

p1=p1->next;

}

while(p2->next!=NULL)

{

fprintf(fp,"%6s %2d %d %d %d\n",p2->sname,p2->sno,p2->chinese,p2->math,p2->english);

p2=p2->next;

}

if(fclose(fp))

{

printf("关闭文件失败");

exit(0);

}

}

#endif // FILEMERGE\_H\_INCLUDED

fileStore补考学生存储

#ifndef FILESTORE\_H\_INCLUDED

#define FILESTORE\_H\_INCLUDED

#include "decsSort.h"

#include "fileMerge.h"

//#include "fileStore.h"

#include "score.h"

#include "search.h"

#include "student.h"

void bu\_kao()//抽取三科成绩中有补考的学生并保存在一个新文件4.txt

{

FILE \*fp;

if((fp=fopen("4.txt","w+"))==NULL){

printf("建立新文件失败");

exit(0);//程序退出

}

gradelist file3=firead("3.txt");//读文件3.txt内容

Student \*p=file3;

printf("补考学生名单");

fprintf(fp,"%s",bt);

while(p->next!=NULL){

if((p->chinese)<60||(p->math)<60||(p->english)<60){

fprintf(fp,"%6s %2d %d %d %d\n",p->sname,p->sno,p->chinese,p->math,p->english);

}

p=p->next;

}

if(fclose(fp)){

printf("不能关闭文件!\n");

exit(0);

}

}

#endif // FILESTORE\_H\_INCLUDED

search查找

#ifndef SEARCH\_H\_INCLUDED

#define SEARCH\_H\_INCLUDED

#include <stdio.h>

#include <stdlib.h>

#include "decsSort.h"

#include "fileMerge.h"

#include "fileStore.h"

#include "score.h"

//#include "search.h"

#include "student.h"

//方法1

void search(char \*name)//按姓名查找

{

gradelist f=firead("3.txt");

Student \*p=f;

while(p->next!=NULL)

{

if(strcmp(name,p->sname)==0)

{

printf("%5s %2d %d %d %d\n",p->sname,p->sno,p->chinese,p->math,p->english);

return;

}

p=p->next;

}

printf("查无此人，请重新输入要查找的信息！\n");

}

//方法2

void search2(char \*name){

gradelist f3=firead("3.txt");

Student \*p=f3;

int n=0;

while(p->next!=NULL){

n++;

p=p->next;

}

typedef struct Stu{

char sname[20];

int sno;

int chinese;

int math;

int english; //新加入总分，方便计算

}gradeFlag;

//排序结构数组，方便输出

typedef struct grade{

gradeFlag g[500];

}gradeTable;

gradeTable T;

p=f3;

int i,flag=0;

for(i=1;i<=n;i++){

//出现数据类型匹配错误

strcpy(T.g[i].sname,p->sname);

T.g[i].sno=p->sno;

T.g[i].chinese=p->chinese;

T.g[i].math=p->math;

T.g[i].english=p->english;

p=p->next;

}

for(i=0;i<n;i++)

{

if(strcmp(T.g[i].sname,name)==0)

{

printf("%5s %2d %d %d %d\n",T.g[i].sname,T.g[i].sno,T.g[i].chinese,T.g[i].math,T.g[i].english);

flag=1;

return;

}

}

if(flag==0){

printf("查无此人，请重新输入要查找的信息！\n");

}

}#endif // SEARCH\_H\_INCLUDED

student学生

#ifndef STUDENT\_H\_INCLUDED

#define STUDENT\_H\_INCLUDED

#include "decsSort.h"

#include "fileMerge.h"

#include "fileStore.h"

#include "score.h"

#include "search.h"

//#include "student.h"

char bt[500];

typedef struct Student{

char sname[20];

int sno;

int chinese;

int math;

int english;

struct Student \*next;

}Student,\*gradelist;

gradelist firead(char \*ad){

FILE \*fp;

//以读文件的方式打开文件

fp=fopen(ad,"r");

//文件为空则退出

if(fp==NULL){

printf("文件打开错误");

exit(0);

}

//为文件申请存储空间

gradelist file=(Student \*)malloc(sizeof(Student));

file ->next =NULL;

Student \*p=file; //Student结构体的指针，方便依次录入数据

int i=0;

//站在光标所在位置,向后看看还有没有字符

while(!feof(fp)){

//i=0时输入表头

if(i==0){

fgets(bt,100,fp);//从指定的流中读取数据，每次读取一行

}

//存入数据之前需要开放空间，动态申请内存空间

if(i==1){

(p->next)=(Student \*)malloc(sizeof(Student));

p=p->next;

p->next=NULL;

}

//存放数据

fscanf(fp,"%s %d %d %d %d",p->sname,&p->sno,&p->chinese,&p->math,&p->english);

i=1;//重新开始存入

}

//关闭文件

if(fclose(fp)){

printf("closs Error!");

exit(0);

}

return file;

}

//将内容显示在屏幕上

void showFile(gradelist file){

//gradelist file=(Student \*)malloc(sizeof(Student));

Student \*p=file;

//将表头打印出来

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

//打印成绩表

while(p->next!=NULL){

printf("%5s %2d %d %d %d\n",p->sname,p->sno,p->chinese,p->math,p->english);

p=p->next;

}

}

#endif // STUDENT\_H\_INCLUDED