# 《问题求解与程序设计》课程实验报告

**学生成绩系统设计与实现**

学 生 姓 名：陈振同

专业、班级 ：软件171

学 号 ：2017211811

联 系 电 话：17861128823

Email ：1765770246@qq.com

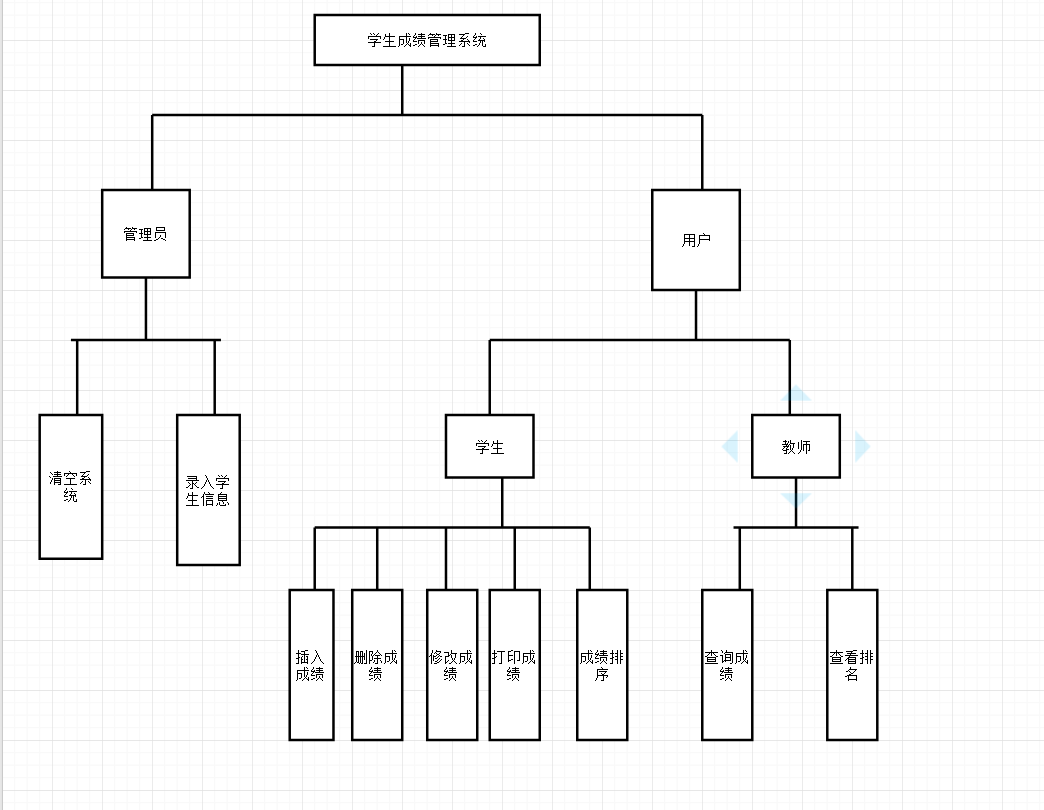
完 成 日 期：2018.4.21

# 设计任务：

随着信息时代的到来，办公趋向无纸化、高效化，对于学生成绩的管理更是如此，基于此，我选择了学生成绩管理系统作为我的课题，希望能够模仿教务系统中成绩的录入与查询功能，编写自己的成绩管理系统。从教师与学生两个角度分别实现成绩的录入、删除、修改、排序、导出与查询功能，学生和教师注册账号后可以完成相应功能。

# 程序的设计与实现：

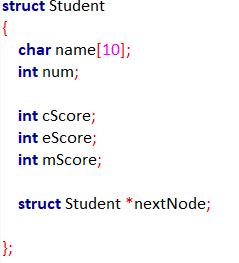
1. 程序结构

本系统可以以管理员和用户身份分别访问。管理员登录时输入系统的默认账号，登录后可以清空系统信息，录入学生信息等.面向两类用户，即教师和学生，用户（学生或教师）可以注册账号，注册后可以进行登录：教师登录后可以对学生信息进行添加成绩、删除成绩、成绩排序、打印成绩、修改成绩等功能，学生登录可进行查询成绩、查看排名等功能。  图一：学生成绩管理系统功能图

# 2.数据描述：

（1）学生信息存储:

本系统运用结构体struct Student存储每个学生的信息，结构体类型定义及结构体成员含义如下。

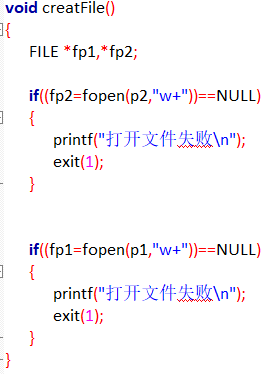


对于班级内学生信息，采用链表的方式存储，链表的头节点声明为:

struct Student \*head=(struct Student\*)malloc(sizeof(struct Student));

（2）账户信息的存储：

系统调用void creatFile()函数可以创建两个文件分别存储学生帐号信息和教师账号信息，用户(学生或教师)注册的信息也将会分别存放两个文件中.



(3)全局变量设计:

定义全局变量char \*p1="asd.txt"（asd.txt存储教师信息）和char \*p2="qwe1.txt"（qwe1.txt存储学生信息）。因为账户的登录分为学生登录和教师登录，学生和教师可以分别登录和注册，而且学生教师的账号信息储存在两个文件中，所以学生和教师注册和登录时只需要传递相应的指针即可打开相应文件完成相应的功能。

定义全局变量stuNum,由管理员向系统输入学生信息

;

# 3.功能模块设计：

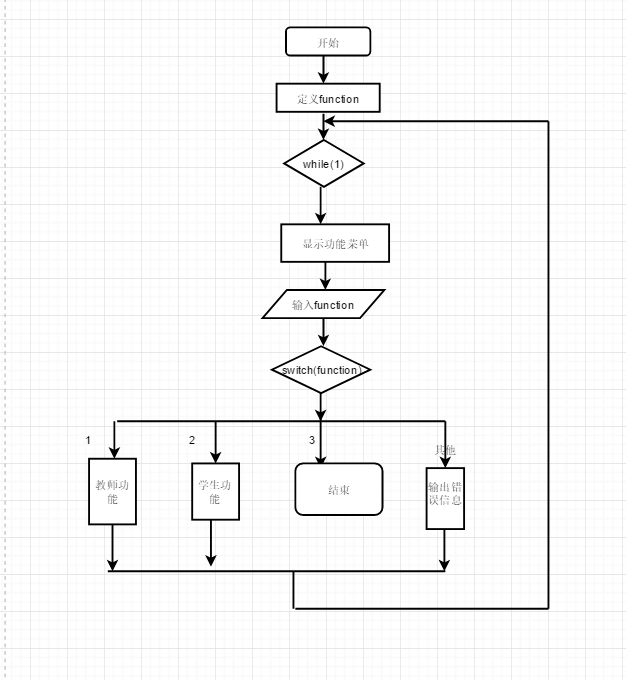
（1）函数名：void menuAll(struct Student \*head)

函数功能：功能选择

输 入：学生结构体头结点,功能（1为教师功能，2为学生功能）

输 出：无

算法描述：见下面流程图



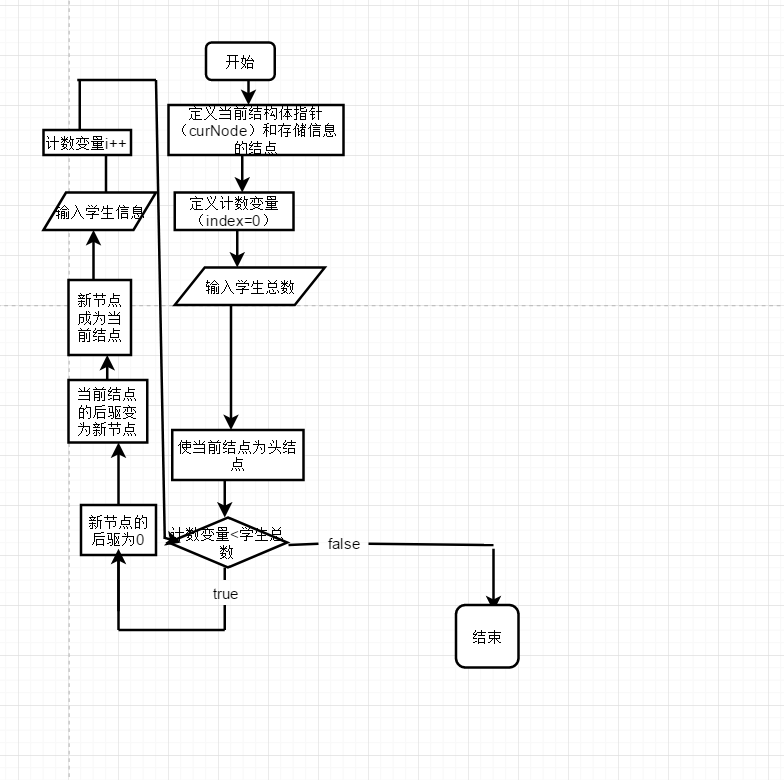
(2)函数名：void creatLinkedList(struct Student \*head)

函数功能：创建一个单链表存储学生信息由管理员输入学生信息

输 入：学生结构体头结点,学生数目及相应的学生信息

输 出：无

算法描述：见下面流程图



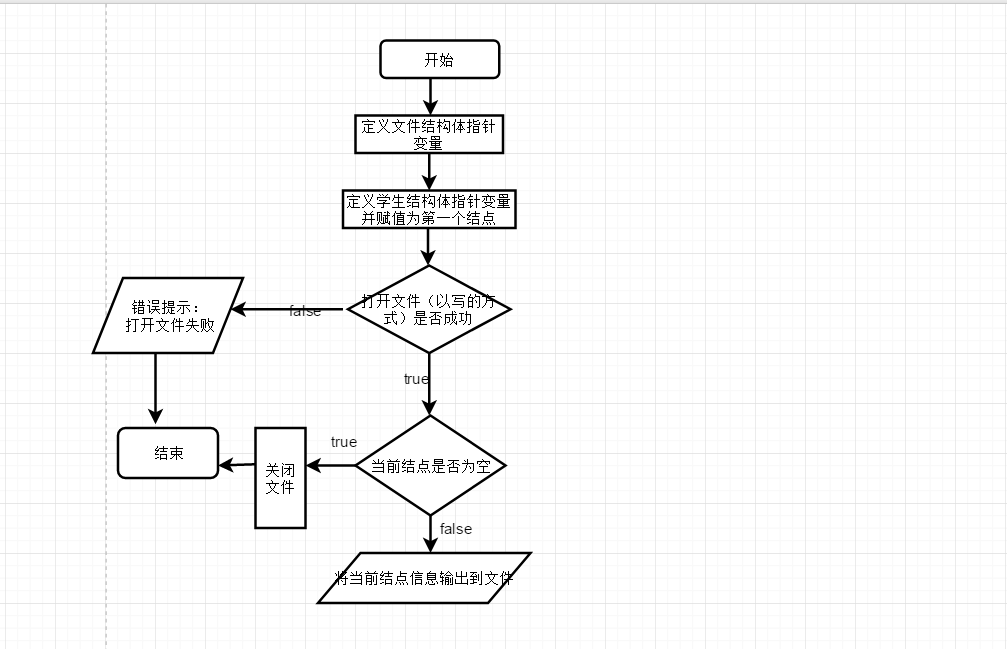
(3)函数名：void saveFile(struct Student \*head)

函数功能：把用链表保存的学生信息写到文件中存储

输 入：学生结构体头结点

输 出：将链表信息读到文件

算法描述：见下面流程图



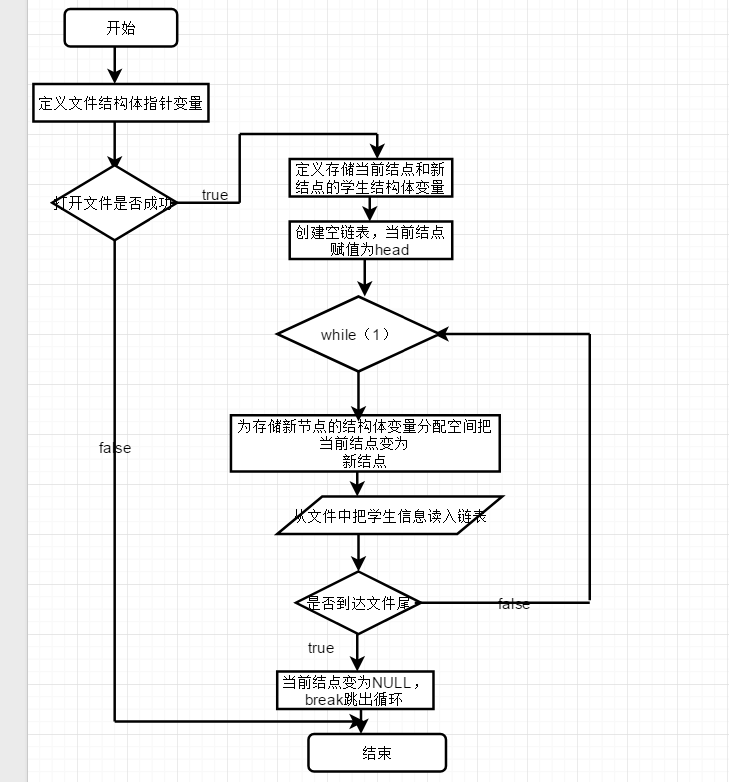
(4)函数名：struct Student\*load(struct Student \*head)

函数功能：从文件中读取学生信息至存储学生信息的链表中

输 入：学生结构体头结点

输 出：学生信息链表的头结点

算法描述：见下面流程图



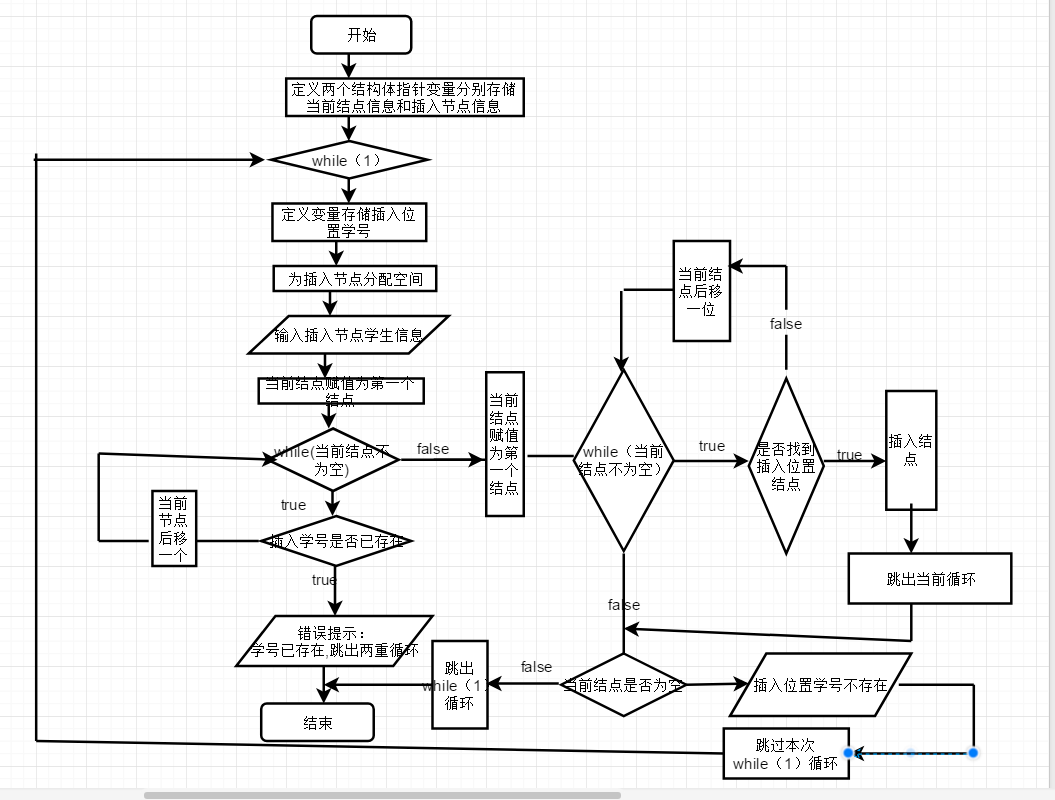
(5)函数名：void insertNode(struct Student \*head)

函数功能：增加一个学生信息

输 入：学生结构体头结点,插入位置学生学号

输 出：无

算法描述：见下面流程图



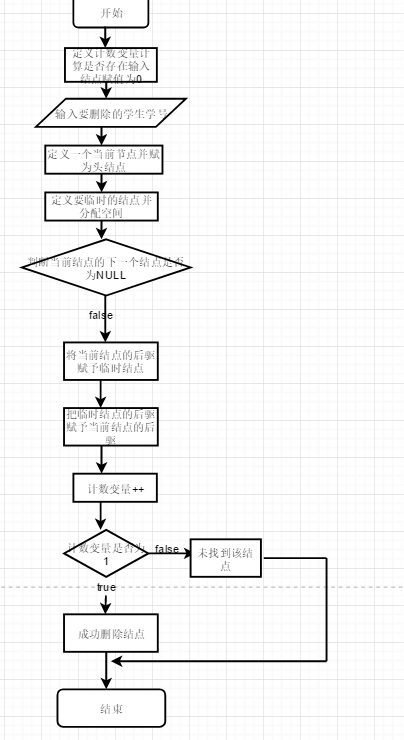
(6)函数名：void deleteNode(struct Student \*head)

函数功能：删除一个学生信息

输 入：学生结构体头结点,删除的学生学号

输 出：无

算法描述：见下面流程图



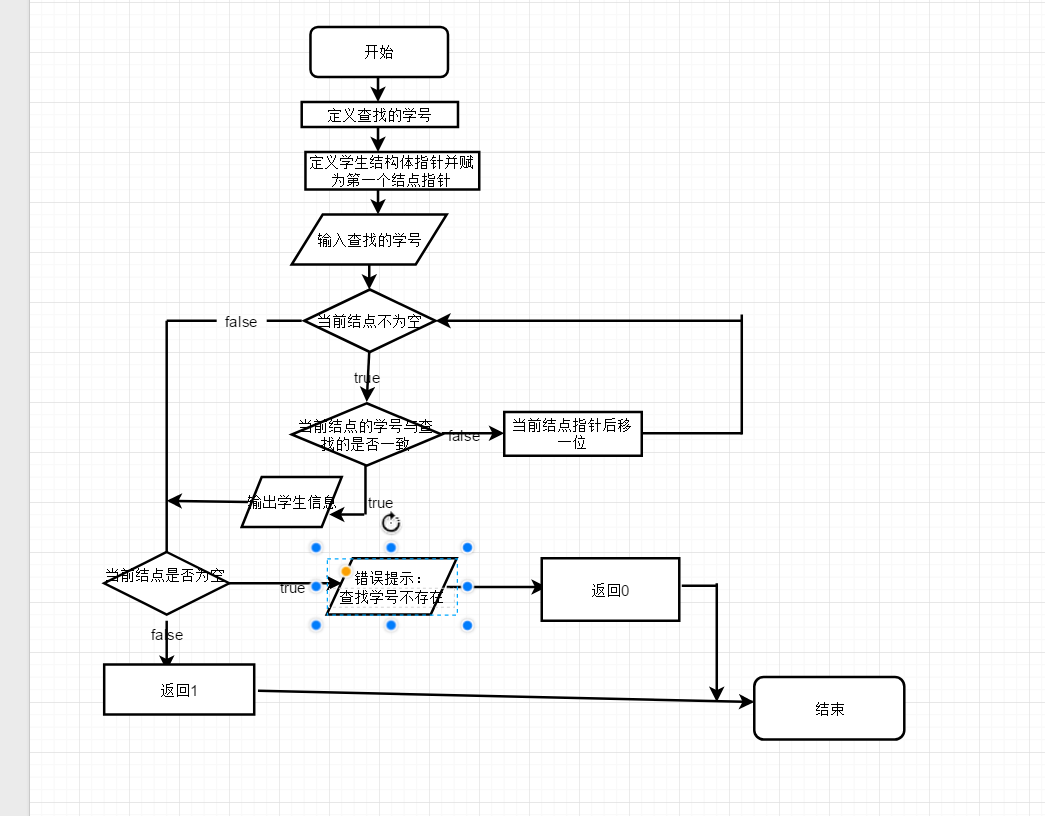
(7)函数名：int findNode(struct Student \*head)

函数功能：查找学生信息

输 入：学生结构体头结点，查找的学生学号

输 出：查找到的学生信息

算法描述：见下面流程图



(8)函数名：void sort(struct Student \*head)

函数功能：对学生总成绩排序

输 入：学生结构体头结点

输 出：排序后学生信息

算法描述：运用选择法排序,交换两个结点的数据域

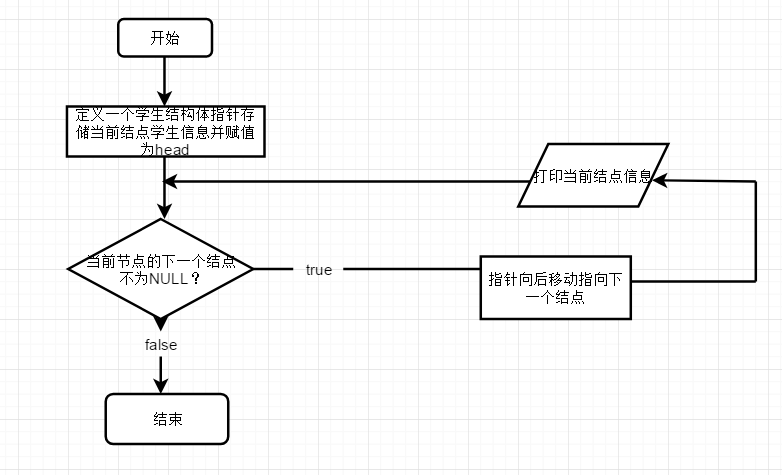
（9）函数名：void print(struct Student \*head)

函数功能：打印学生信息

输 入：学生结构体头结点

输 出：无

算法描述：见下面流程图



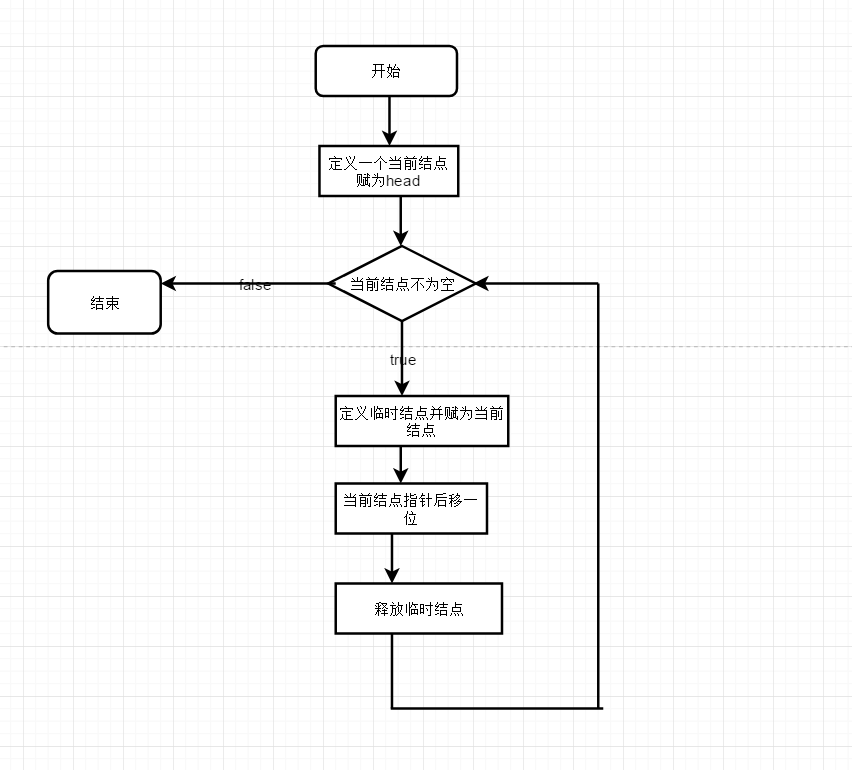
（10）函数名：void freeNode(struct Student \*head)

函数功能：释放所有节点

输 入：学生结构体头结点

输 出：无

算法描述：无



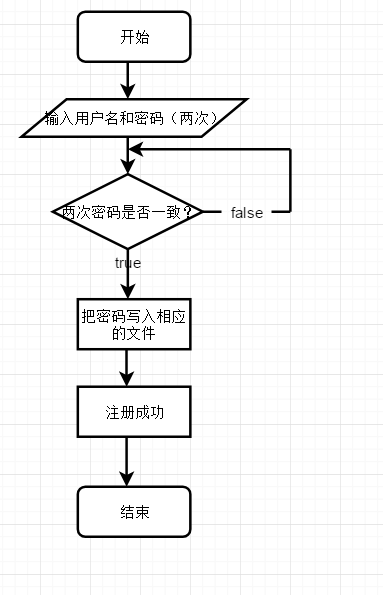
（9）函数名：void regist(char \*p)

函数功能：注册一个账号（学生或教师）

输 入：用户名与密码（输入两次）

输 出：无

算法描述：将注册的账号写入教师学生对应的文件中



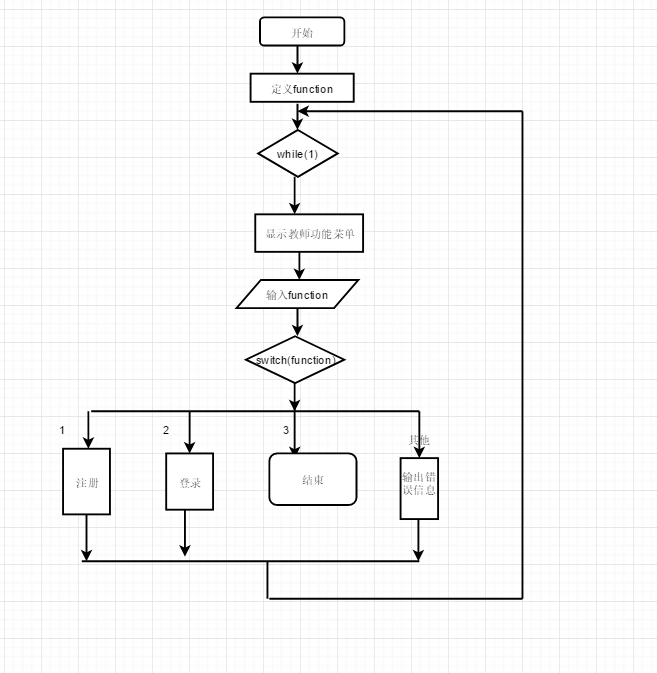
（10）函数名：void teachFunction(struct Student\*head)

函数功能:教师账号注册或登录

输 入：学生结构体头结点，功能（注册或登录）

输 出：无

算法描述：见下面流程图



（11）

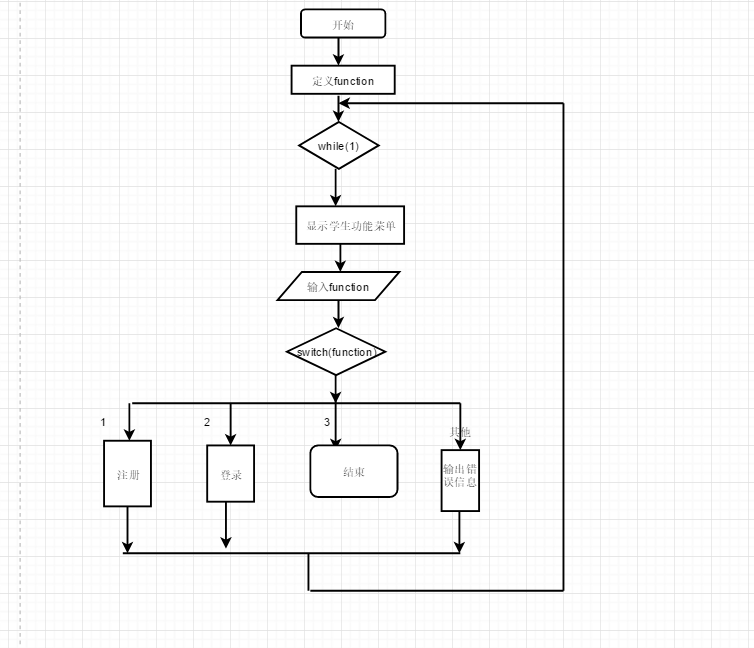
函数名：void stuFunction(struct Student\*head)

函数功能:教师账号注册或登录

输 入：学生结构体头结点,功能（注册或登录）

输 出：无

算法描述：见下面流程图



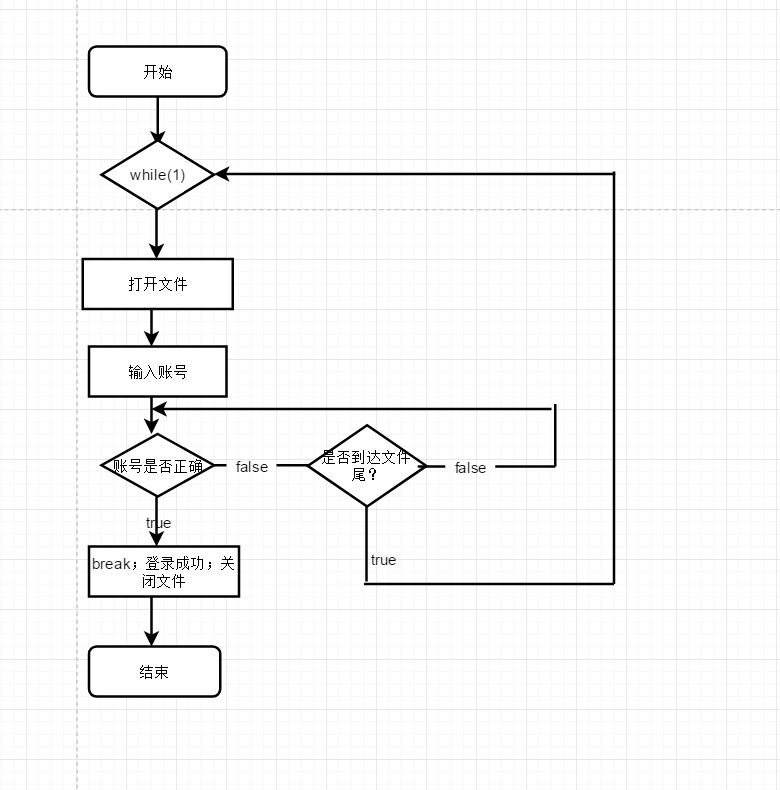
（12）函数名：void passWords(char \*p)

函数功能：输入密码并判断密码是否正确

输 入：用户名和密码

输 出：无

算法描述：见以下流程图



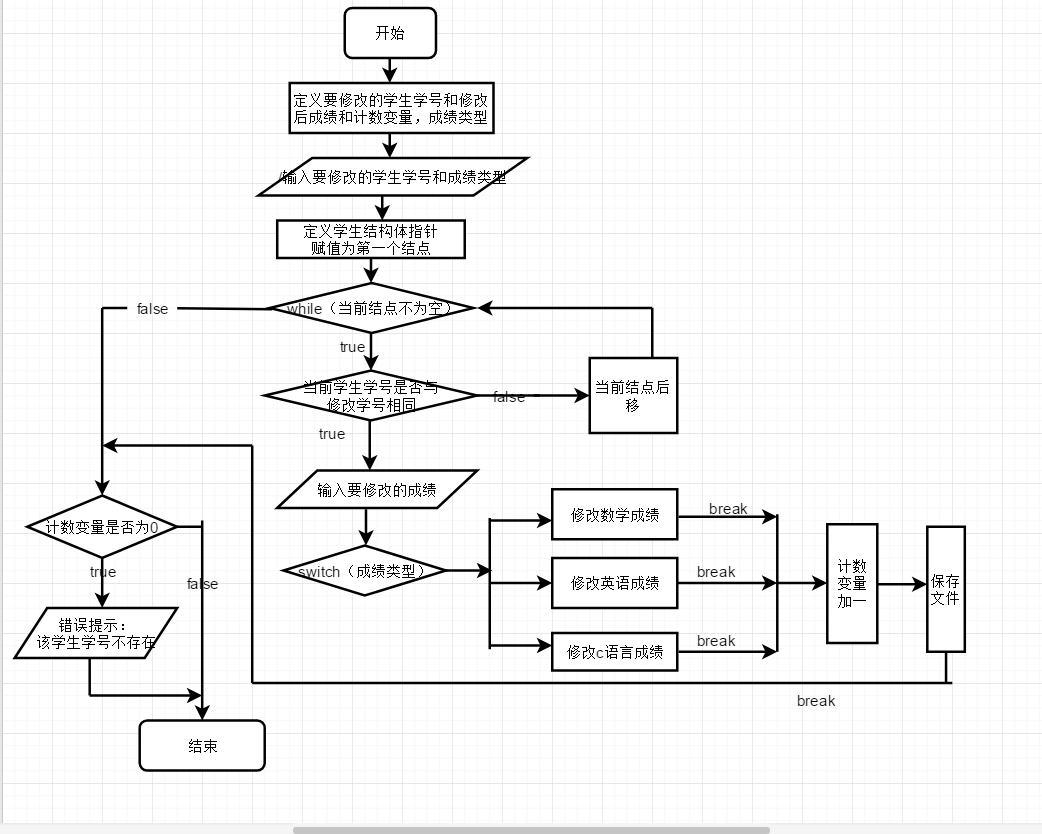
（13）函数名：void changeScore(struct Student\*head)

函数功能：修改 成绩

输 入：学生结构体头结点、要修改的科目、学生学号和要修改的成绩

输 出：无

算法描述：见以下流程图



（14）函数名：void clearInformation()

函数功能：清空学生信息和学生教师账号的全部信息

输 入：无

输 出：无

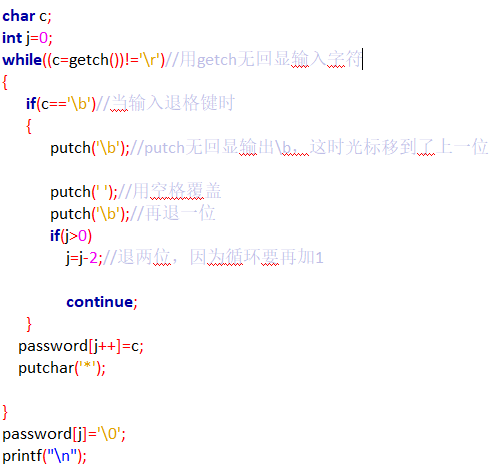
算法描述：以写的方式打开三个文件

.

# 设计难点与解决方法

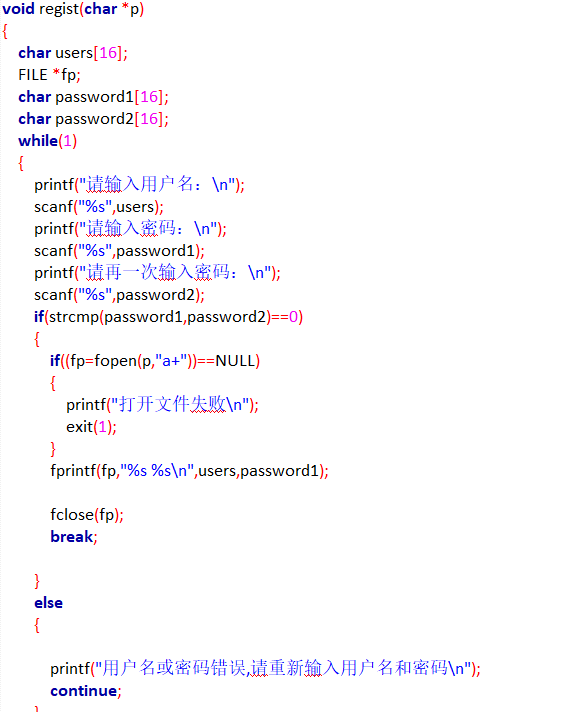
。

3.1密码输入时隐藏以及密码回删功能



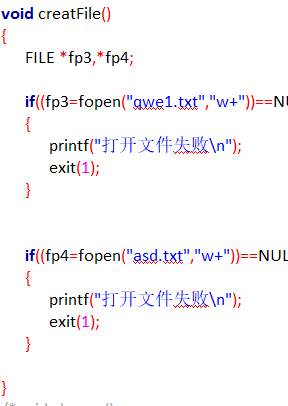
3.2教师和学生账号的注册

从键盘输入用户名密码，用户名与密码分别保存于教师和学生存储账号的文件中



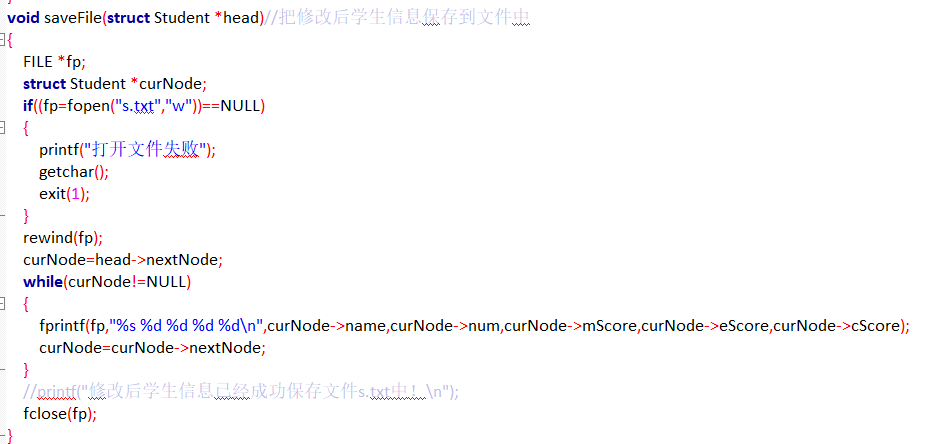
3.3学生和教师账号的储存

写两个文件分别存储学生帐号和教师账号



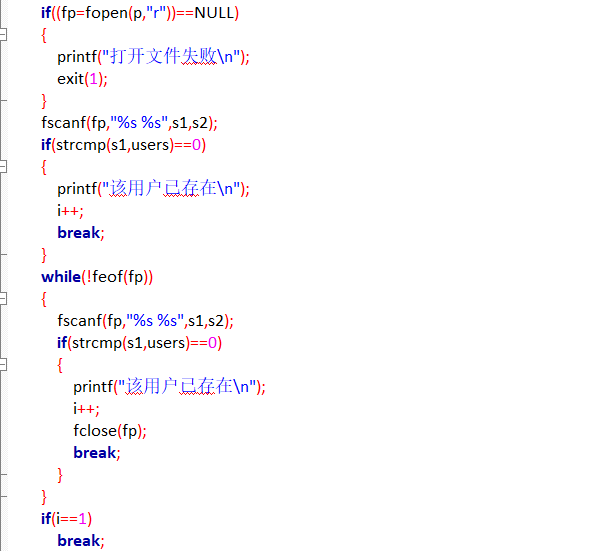
3.4修改数据后在文件中数据未改变

解决方法：每次修改数据，都将数据保存到文件中



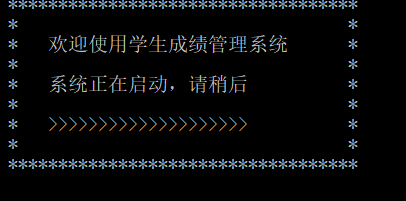
3.5注册账号时如果用户名已经存在的问题

解决方法：读取文件，判定注册的账号是否存在

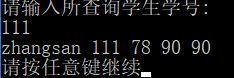


# 设计效果与不足之处

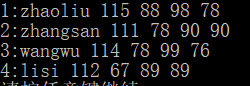
4.1. 系统启动界面，调用system（“cls”）;Sleep();



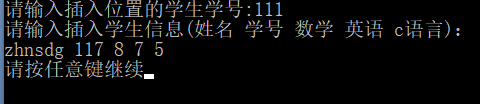
4.2学生能够学号查询自己的成绩

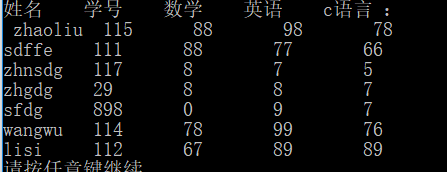


4.3学生登录后可查询班级成绩表

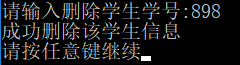


4.4教师登录后可以增加学生信息

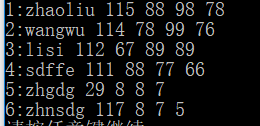




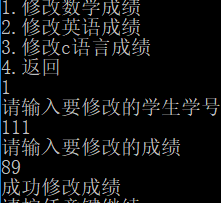
4.4教师可以删除学生信息



4.5教师可以学生成绩排序



4.6教师可以修改学生成绩



4.6.不足之处：

（1）界面不够优美，数据输出不整齐

（2）不能完全排除非法输入

（3）系统功能有待完善，许多功能尚未实现

（4）有些算法不够简便，有待提高

（5）对于函数模块的划分不够详细，一个函数应该只实现一个功能

（6）有些函数比较类似，可以变为一个函数，

# 总结

通过此次实验项目，我学到了：

（1）能够熟练的应用各种循环语句和选择语句

（2）通过实现系统的增加，删除，查找，排序等功能，加强了对于链表基本操作的学习

（3）将数据写入文件，用文件读写数据，加强了对于文件的学习和理解

（4）实现每个部分的功能，能够更加独立的思考，分析解决问题

（5）注释的重要性

自己在做项目时，存在哪些问题：

（1）遇到问题时，不能清晰地了解问题的根源，往往一个问题要拖好几天才能解决

（2）对于c语言基础知识的熟练程度有待提高

（3）遇到困难，不够耐心，信心不足

(4) 对于一个程序实现，自己想的总是实现就好，而不去考虑简便算法

后续如何改正：

1. 遇到问题，在可能出现bug的地方设置断点调试，寻找问题的根源所在
2. 今后多做项目，敲代码，加强对于自己知识的应用
3. 要勇敢地面对困难，在困难面前不要低头
4. 加强数学等与计算机相关课程的学习，增强自己分析解决问题的能力

# 附件

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<windows.h>

#include<conio.h>

#define DEBUG 1

struct Student

{

char name[10];

int num;

int cScore;

int eScore;

int mScore;

struct Student \*nextNode;

};

char \*p1="teacher.txt";//存储教师账号

char \*p2="student.txt";//存储学生账号

char \*p3="s.txt";//存储学生信息

char mangers[10]="zhangbin";

char mangerPassword[10]="123456";

void identityView(struct Student\*head);//以何身份访问

void manger(struct Student\*head);//管理员功能

void clearInformation();//清除信息

void changeScore(struct Student\*head,int choice);//修改成绩

void changeSelect(struct Student\*head);//选择修改成绩的种类

void launchBoundar();//启动界面

void menuAll(struct Student \*head);//功能菜单

void menus();//功能界面

void Menu2();//登录注册界面

void teachFunction(struct Student\*head);//教师功能

void stuFunction(struct Student\*head);//学生功能

void stuLogin(struct Student\*head);//学生登录

void teachLogin(struct Student\*head);//教师登录

void regist(char \*p);//注册

void mangerMenu();//管理员菜单

void passWords(char \*p);//账号输入

void stuFunctionMenu();//学生功能菜单

void aRegist();//注册

void loginMenu();//登录菜单

void anyKeyGoOn();//按任意键继续

void mainMenu();//主菜单

void mangerMenus();//管理员菜单

void login(struct Student \*head);//登录

void creatLinkedList(struct Student \*head);//创建链表由管理员输入学生数目和信息

struct Student\*load(struct Student \*head);//从文件中读取学生信息

void saveFile(struct Student \*head);//存储修改后学生信息至文件

void sort(struct Student \*head);//学生成绩排序

void deleteNode(struct Student \*head);//删除学生信息

void print(struct Student \*head);//打印学生信息

void freeNode(struct Student \*head);//释放链表

int findNode(struct Student \*head);//查找学生信息

void insertNode(struct Student \*head);//插入学生信息

int main()

{

struct Student \*head=(struct Student \*)malloc(sizeof(struct Student));

load(head);

system("cls");

launchBoundar();

Sleep(2000);

identityView(head);

freeNode(head); //释放链表

return 0;

}

void creatLinkedList(struct Student \*head)

{

struct Student \*curNode,\*newNode;

int index=0;

int studentNum;

printf("请输入学生数目:");

scanf("%d",&studentNum);

printf("请输入学生信息(姓名，学号，数学，英语，c语言)：\n");

curNode=head;

while(index<studentNum)

{

newNode=(struct Student \*)malloc(sizeof(struct Student));

newNode->nextNode=NULL;

curNode->nextNode=newNode;

curNode=newNode;

scanf("%s %d %d %d %d",newNode->name,&newNode->num,&newNode->mScore,&newNode->eScore,&newNode->cScore);

index++;

}

saveFile(head);

}

void print(struct Student \*head)

{

struct Student \*curNode=head;

printf("姓名 学号 数学 英语 c语言 ：\n ");

while(curNode->nextNode!=NULL)

{

curNode=curNode->nextNode;

printf("%-8s %-8d %-8d %-8d %-8d\n",curNode->name,curNode->num,curNode->mScore,curNode->eScore,curNode->cScore);

}

}

//释放链表节点

void freeNode(struct Student \*head)

{

struct Student \*curNode;

curNode=head;

while(curNode)

{

struct Student \*temp;

temp=curNode;

curNode=curNode->nextNode;

free(temp);

}

}

//删除学生信息

void deleteNode(struct Student \*head)

{

int numDelete,i=0;

printf("请输入删除学生学号:");

scanf("%d",&numDelete);

struct Student \*deleteNode;

struct Student \*curNode;

deleteNode=(struct Student\*)malloc(sizeof(struct Student));

curNode=head;

while(curNode->nextNode!=NULL)

{

if(curNode->nextNode->num==numDelete)

{

deleteNode=curNode->nextNode;

curNode->nextNode=deleteNode->nextNode;

free(deleteNode);

i++;

break;

}

curNode=curNode->nextNode;

}

if(i==0)

{

printf("该学号不存在 \n");

}

else

{

printf("成功删除该学生信息\n");

}

saveFile(head);

}

//查找学生信息

int findNode(struct Student \*head)

{

int findNum;

printf("请输入所查询学生学号:\n");

scanf("%d",&findNum);

struct Student \*curNode;

curNode=head->nextNode;

while(curNode!=NULL)

{

if(curNode->num==findNum)

{

printf("%s %d %d %d %d\n",curNode->name,curNode->num,curNode->mScore,curNode->eScore,curNode->cScore);

break;

}

curNode=curNode->nextNode;

}

if(curNode!=NULL)

{

return 1;

}

else

{

printf("该学生学号不存在\n");

return 0;

}

}

//在某学生后面增加学生信息

void insertNode(struct Student \*head)

{

struct Student \*pNew;

struct Student \*curNode;

while(1)

{

int numInsert;

printf("请输入插入位置的学生学号:");

scanf("%d",&numInsert);

printf("请输入插入学生信息(姓名 学号 数学 英语 c语言)：\n");

pNew=(struct Student\*)malloc(sizeof(struct Student));

scanf("%s %d %d %d %d",pNew->name,&pNew->num,&pNew->mScore,&pNew->eScore,&pNew->cScore);

curNode=head->nextNode;

while(curNode!=NULL)

{

if(curNode->num==pNew->num)

{

printf("所插入的学生学号已存在");

break;

break;

}

curNode=curNode->nextNode;

}

curNode=head->nextNode;

while(curNode!=NULL)

{

if(curNode->num==numInsert)

{

pNew->nextNode=curNode->nextNode;

curNode->nextNode=pNew;

break;

}

curNode=curNode->nextNode;

}

if(curNode!=NULL)

{

break;

}

else

{

printf("您所插入位置的学生学号不存在\n");

continue;

}

}

saveFile(head);

}

void changeSelect(struct Student\*head)

{

int choice;

while(1)

{

printf("1.修改数学成绩\n");

printf("2.修改英语成绩\n");

printf("3.修改c语言成绩\n");

printf("4.返回\n");

scanf("%d",&choice);

switch(choice)

{

case 1:

changeScore(head,choice);//修改数学成绩

break;

case 2:

changeScore(head,choice);//修改英语成绩

break;

case 3:

changeScore(head,choice);//修改c语言成绩

break;

case 4:

break;

default:

printf("输入有误");

Sleep(2000);

system("cls");

continue;

}

break;

}

}

void changeScore(struct Student\*head,int choice)

{

int id,score,i=0;

struct Student\*curNode;

while(1)

{

printf("请输入要修改的学生学号\n");

scanf("%d",&id);//输入要修改的学生学号

curNode=head->nextNode;

while(curNode)

{

if(curNode->num==id)

{

printf("请输入要修改的成绩\n");

scanf("%d",&score);

switch(choice)

{

case 1:

curNode->mScore=score;

break;

case 2:

curNode->eScore=score;

break;

case 3:

curNode->cScore=score;

break;

}

i++;

saveFile(head);

printf("成功修改成绩\n");

break;

}

curNode=curNode->nextNode;

}

if(i==0)

{

printf("该学生学号不存在 \n");

Sleep(2000);

//anyKeyGoOn();

system("cls");

break;

}

break;

}

}

void sort(struct Student \*head)//选择法排序

{

int i=0,a[4]= {0},b[4]= {0};

char s[10];

struct Student\*sortNode=(struct Student\*)malloc(sizeof(struct Student));

struct Student \*curNode;

for(curNode=head->nextNode; curNode!=NULL; curNode=curNode->nextNode)

{

for(sortNode=curNode->nextNode; sortNode!=NULL; sortNode=sortNode->nextNode)

{

b[0]=sortNode->cScore;

b[1]=sortNode->eScore;

b[2]=sortNode->mScore;

b[3]=sortNode->num;

a[0]= curNode->cScore;

a[1]=curNode->eScore;

a[2]=curNode->mScore;

a[3]=curNode->num;

strcpy(s,sortNode->name);

if(a[0]+a[1]+a[2]<b[0]+b[1]+b[2])

{

curNode->cScore=b[0];

curNode->eScore=b[1];

curNode->mScore=b[2];

curNode->num=b[3];

sortNode->cScore=a[0];

sortNode->eScore=a[1];

sortNode->mScore=a[2];

sortNode->num=a[3];

strcpy(sortNode->name,curNode->name);

strcpy(curNode->name,s);

}

}

}

curNode=head;

i=1;

while(curNode->nextNode!=NULL)

{

curNode=curNode->nextNode;

printf("%d:%s %d %d %d %d\n",i,curNode->name,curNode->num,curNode->mScore,curNode->eScore,curNode->cScore);

i++;

}

}

void saveFile(struct Student \*head)//把修改后学生信息保存到文件中

{

FILE \*fp;

struct Student \*curNode;

if((fp=fopen(p3,"w"))==NULL)

{

printf("打开文件失败");

getchar();

exit(1);

}

rewind(fp);

curNode=head->nextNode;

while(curNode!=NULL)

{

fprintf(fp,"%s %d %d %d %d\n",curNode->name,curNode->num,curNode->mScore,curNode->eScore,curNode->cScore);

curNode=curNode->nextNode;

}

//printf("修改后学生信息已经成功保存文件s.txt中！\n");

fclose(fp);

}

struct Student\*load(struct Student \*head)//从文件中读取学生信息

{

struct Student \*curNode;

int i=0;

FILE \*fp;

if((fp=fopen(p3,"r"))==NULL)

{

printf("打开文件失败");

getchar();

}

head->nextNode=NULL;//创建空链表

curNode=head;

struct Student \*newNode;

while(1)

{

newNode=(struct Student\*)malloc(sizeof(struct Student));

newNode->nextNode=NULL;

curNode->nextNode=newNode;

curNode=newNode;

fscanf(fp,"%s%d%d%d%d\n",newNode->name,&newNode->num,&newNode->mScore,&newNode->eScore,&newNode->cScore);

i++;

printf("已经读入数据%d条\n",i);

if(feof(fp))

{

curNode=NULL;

break;

}

}

fclose(fp);

return head;

}

void passWords(char \*p)//登录账户密码的输入（隐藏密码）

{

char s1[17],s2[17];

FILE \*fp;

char user\_name[17];

int i=0;

char password[17];

if((fp=fopen(p,"r+"))==NULL)

{

printf("打开文件失败\n");

exit(1);

}

fscanf(fp,"%s %s",s1,s2);

while(!feof(fp))

{

fscanf(fp,"%s %s",s1,s2);

}

fclose(fp);

while(1)

{

if((fp=fopen(p,"r+"))==NULL)

{

printf("打开文件失败\n");

exit(1);

}

fscanf(fp,"%s %s",s1,s2);

printf("请输入您的用户名：\n");

scanf("%s",user\_name);

printf("请输入您的密码：\n");

char c;

int j=0;

while((c=getch())!='\r')//用getch无回显输入字符

{

if(c=='\b')//当输入退格键时

{

putch('\b');//putch无回显输出\b，这时光标移到了上一位

putch(' ');//用空格覆盖

putch('\b');//再退一位

if(j>0)

j=j-2;//退两位，因为循环要再加1

continue;

}

password[j++]=c;

putchar('\*');

}

password[j]='\0';

printf("\n");

if((strcmp(s1,user\_name)==0&&strcmp(s2,password)==0))

{

printf("登录成功");

i++;

break;

}

while(!feof(fp))

{

fscanf(fp,"%s %s",s1,s2);

if((strcmp(s1,user\_name)==0&&strcmp(s2,password)==0))

{

printf("登录成功");

i++;

break;

}

}

if(i==0)

{

fclose(fp);

printf("用户名或密码错误，请重新输入\n");

continue;

}

if(i==1)

{

fclose(fp);

break;

}

}

}

void regist(char \*p)//账号注册

{

char users[17];

FILE \*fp;

int i=0;

char s1[17],s2[17];

char password1[17];

char password2[17];

while(1)

{

i=0;

printf("\t\t\t\t\t\t账号注册\n");

printf("\t\t\t\t\t用户名:");

scanf("%s",users);

printf("\t\t\t\t\t密码:");

scanf("%s",password1);

printf("\t\t\t\t\t确认密码:");

scanf("%s",password2);

if((fp=fopen(p,"r"))==NULL)

{

printf("打开文件失败\n");

exit(1);

}

fscanf(fp,"%s %s",s1,s2);

if(strcmp(s1,users)==0)

{

printf("该用户已存在\n");

i++;

break;

}

while(!feof(fp))

{

fscanf(fp,"%s %s",s1,s2);

if(strcmp(s1,users)==0)

{

printf("该用户已存在\n");

i++;

fclose(fp);

break;

}

}

if(i==1)

break;

if(strcmp(password1,password2)==0)

{

if((fp=fopen(p,"a+"))==NULL)

{

printf("打开文件失败\n");

exit(1);

}

fprintf(fp,"%s %s\n",users,password1);

printf("\t\t\t\t\t注册成功！\n");

Sleep(2000);

system("cls");

fclose(fp);

break;

}

else

{

printf("\t\t\t\t\t两次密码不一致,请重新输入\n");

//Sleep(2000);

continue;

}

}

}

void clearInformation()

{

FILE \*fp1,\*fp2,\*fp;

if((fp2=fopen(p2,"w"))==NULL)

{

printf("打开文件失败\n");

exit(1);

}

if((fp1=fopen(p1,"w"))==NULL)

{

printf("打开文件失败\n");

exit(1);

}

if((fp=fopen(p3,"w"))==NULL)

{

printf("打开文件失败\n");

exit(1);

}

}

void teachFunctionMenu();

void teachLogin(struct Student\*head)

{

passWords(p1);

Sleep(2000);

system("cls");

int choice;

while(1)

{

teachFunctionMenu();

scanf("%d",&choice);

switch(choice)

{

case 1:

insertNode(head);//增加学生信息

anyKeyGoOn();

system("cls");

continue;

case 2:

deleteNode(head);//删除学生信息

anyKeyGoOn();

system("cls");

continue;

case 3:

sort(head);//对学生成绩排序

anyKeyGoOn();

system("cls");

continue;

case 4:

print(head);//打印学生成绩

anyKeyGoOn();

system("cls");

continue;

case 5:

changeSelect(head);//修改学生成绩

anyKeyGoOn();

system("cls");

continue;

case 6:

break;

default:

printf("\t\t\t\t输入错误,请重新输入\n");

Sleep(2000);

continue;

}

system("cls");

break;

}

}

void teachFunctionMenu()

{

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t 学生成绩管理系统教师端\n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_1.增加学生信息\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_2.删除学生信息\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_3.学生成绩排序\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_4.打印学生成绩\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_5.修改学生成绩\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_6.返回\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

void stuFunctionMenu()

{

printf("\t\t\t\t\t 学生成绩管理系统学生端\n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_1.查询个人成绩\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_2.查看班级排名\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_3.返回\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

void stuLogin(struct Student\*head)

{

passWords(p2);

Sleep(2000);

system("cls");

int choice;

while(1)

{

stuFunctionMenu();

scanf("%d",&choice);

switch(choice)

{

case 1:

findNode(head);//增加学生信息

anyKeyGoOn();

system("cls");

continue;

case 2:

sort(head);//删除学生信息

anyKeyGoOn();

system("cls");

continue;

case 3:

break;

default:

printf("输入错误,请重新输入\n");

Sleep(2000);

system("cls");

continue;

}

system("cls");

break;

}

}

void anyKeyGoOn()

{

char ch;

printf("请按任意键继续");

ch=getch();

}

void stuLogin(struct Student \*head);

void regist(char \*p);

void stuFunction(struct Student\*head)

{

int function;

while(1)

{

printf("\t\t\t\t\t\t\t学生端\n");

Menu2();

scanf("%d",&function);

switch(function)

{

case 1:

system("cls");

printf("\t\t\t\t\t\t学生帐号注册\n");

regist(p2);

anyKeyGoOn();

system("cls");

continue;

case 2:

system("cls");

printf("\t\t\t\t\t\t学生帐号登录\n");

stuLogin(head);

anyKeyGoOn();

system("cls");

continue;

case 3:

break;

default:

anyKeyGoOn();

system("cls");

continue;

}

system("cls");

break;

}

}

void teachLogin(struct Student\*head);

void regist(char \*p);

void teachFunction(struct Student\*head)

{

int function;

while(1)

{

printf("\t\t\t\t\t\t\t教师端\n");

Menu2();

scanf("%d",&function);

switch(function)

{

case 1:

system("cls");

printf("\t\t\t\t\t\t教师帐号注册\n");

regist(p1);

anyKeyGoOn();

system("cls");

continue;

case 2:

system("cls");

printf("\t\t\t\t\t\t教师帐号登录\n");

teachLogin(head);

anyKeyGoOn();

system("cls");

continue;

case 3:

break;

default:

printf("输入错误，请您重新输入\n");

Sleep(2000);

system("cls");

continue;

}

system("cls");

break;

}

}

void Menus()

{

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t 学生成绩管理系统\n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1.教师端\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2.学生端\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3.返回\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

void Menus();

void teachFunction(struct Student\*head);

void stuFunction(struct Student\*head);

void menuAll(struct Student \*head)

{

int function;

while(1)

{

Menus();

scanf("%d",&function);

switch(function)

{

case 1:

system("cls");

teachFunction(head);

anyKeyGoOn();

system("cls");

continue;

case 2:

system("cls");

stuFunction(head);

anyKeyGoOn();

system("cls");

continue;

case 3:

break;

default:

printf("输入错误，请重新输入\n");

Sleep(2000);

system("cls");

continue;

}

system("cls");

break;

}

}

void Menu2()

{

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t 学生成绩管理系统\n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1.注册\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2.登录\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3.返回\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

void launchBoundar()

{

printf("\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\t\t\* \*\n");

printf("\t\t\t\t\* 欢迎使用学生成绩管理系统 \*\n");

printf("\t\t\t\t\* \*\n");

printf("\t\t\t\t\* 系统正在启动，请稍后 \*\n");

printf("\t\t\t\t\* \*\n");

printf("\t\t\t\t\* >>>>>>>>>>>>>>>>>>>> \*\n");

printf("\t\t\t\t\* \*\n");

printf("\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Sleep(2000);

system("cls");

}

void identityView(struct Student\*head)

{

int identity;

while(1)

{

mainMenu();

scanf("%d",&identity);

switch(identity)

{

case 1:

system("cls");

manger(head);

anyKeyGoOn();

system("cls");

continue;

case 2:

system("cls");

menuAll(head);

anyKeyGoOn();

system("cls");

continue;

case 3:

break;

default:

printf("输入错误\n");

Sleep(2000);

system("cls");

continue;

}

system("cls");

break;

}

}

void manger(struct Student\*head)

{

char s1[10],s2[10];

int choice;

while(1)

{

printf("输入用户名和密码\n");

scanf("%s %s",s1,s2);

if(strcmp(s1,mangers)==0&&strcmp(s2,mangerPassword)==0)

{

printf("登录成功\n");

break;

}

else

{

printf("用户名或密码错误\n");

continue;

}

}

while(1)

{

mangerMenus();

scanf("%d",&choice);

switch(choice)

{

case 1:

system("cls");

clearInformation();//清空系统

anyKeyGoOn();

system("cls");

continue;

case 2:

system("cls");

creatLinkedList(head);//加入学生信息

anyKeyGoOn();

system("cls");

continue;

case 3:

break;

default:

printf("输入错误\n");

Sleep(2000);

system("cls");

continue;

}

break;

}

}

void mainMenu()

{

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t 学生成绩管理系统\n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1.管理员访问\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2.用户访问\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3.退出系统\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

void mangerMenus()

{

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t 学生成绩管理系统\n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1.清空系统\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2.加入学生信息\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3.返回\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\t\t\t\t\t \n");

printf("\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}