## 第12章 综合实例——学生信息管理程序

学生信息管理系统是一个信息化管理软件，可以帮助学校快速录入学生的信息，并且对学生的信息进行基本的增删改操作还可以根据排序功能，宏观地看到学生成绩从高到低的排列，随时掌握学生的学习状态，并能实时地将学生的信息保存到磁盘文件中，方便查看。

## 12.1开发背景

在科技日益发展的今天，学生成为了国家关注培养的重点，衡量一个学生在校状态的指标就是学生的成绩现如今的学生多，信息更新快，手工记录的学生信息已经跟不上时代的发展，容易出错，不能及时反应给家长老师和同学关于学生成绩的更新，对学生最近的状态不能很快地定位。而智能化、信息化的学生信息管理系统更方便快捷地统计学生的信息，记录学生的信息，对学生信息的变化及更新，同样可以使人们实时地了解学生的成绩动态，更好地管理学生。

## 12.2开发环境

本项目的开发环境如下：

操作系统：Windows 7。

开发工具：Visual C++6.0。

开发语言：C语言。

## 12.3 系统功能设计

根据分析，可将系统分为7大功能模块，主要包括输入记录模块、查找记录模块、删除记录模块、修改记录模块、插入记录模块、排序模块、统计个数模块、显示记录模块等。系统的主要功能结构图如图12-1所示。

图12-1 系统功能结构图

## 12.4预处理模块和结构体

学生信息管理系统在预处理模块中宏定义了在整个系统程序中常用到的结构体类型的长度，以及输入输出的格式说明，由于在学生信息的结构体中成员太多，对所有的成员进行应用时，代码太过于冗长、输入容易出错，所以在预处理模块中将其宏定义为DATA。预处理模块还对系统中的各个功能模块的函数做了声明。

另外，由于学生信息的数据较多，信息数据类型各不相同，显示学生成员息时会比较凌乱，为了使界面简洁美观，本案例使用了format语句对输出的格式说明进行规划。可以用如下代码解决:

#define FORMAT "%-12d%-12s%-12.1f%-12.1f%-12.1f%-12.1f\n"

以上代码对输出的格式控制部分进行宏定义，每一个格式说明中间都插有附加字符。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*预处理模块和结构体\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<dos.h>

#include<string.h>

#define LEN sizeof(struct student)

#define FORMAT "%-12d%-12s%-12.1f%-12.1f%-12.1f%-12.1f\n"

#define DATA stu[i].num,stu[i].name,stu[i].chinese,stu[i].math,stu[i].english,stu[i].sum

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*定义学生成绩结构体\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

struct student

{

int num; /\*学号\*/

char name[15]; /\*姓名\*/

float chinese; /\*语文课\*/

float math; /\*数学课\*/

float english; /\*英语课\*/

float sum; /\*总分\*/

};

struct student stu[50]; /\*定义全局结构体数组\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*函数声明与简介\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void input(); /\*录入学生成绩信息\*/

void show(); /\*显示学生基本信息\*/

void order(); /\*按总分排序\*/

void del(); /\*删除学生成绩信息\*/

void modify(); /\*修改学生成绩信息\*/

void menu(); /\*程序主菜单\*/

void insert(); /\*插入学生信息\*/

void total(); /\*计算总人数\*/

void search(); /\*查找学生信息\*/

## 12.5函数设计

### 12.5.1 主函数

运行学生信息管理系统，第一步会看到一个主功能菜单以供用户选择，在菜单中列出本系统中的所有功能，以及如何调用相应的功能，程序对相应的功能进行了编号，用户可以根据需要输入对应的功能编号来调用子函数。其中menu( )函数主要使用printf( )函数输出文字等信息。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*主函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

int n;

menu();

scanf("%d",&n);/\*输入选择功能的编号\*/

while(n)

{

switch(n)

{

case 0: exit(0);

case 1: input();break;

case 2: search();break;

case 3: del();break;

case 4: modify();break;

case 5: insert();break;

case 6: order();break;

case 7: total();break;

case 8: show();break;

default:printf("输入有误，请重新输入:");break;

}

getch();

menu(); /\*菜单界面循环显示\*/

scanf("%d",&n);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*自定义函数实现菜单功能\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void menu()

{

system("cls");

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

printf("\t\t|-------------学生信息管理系统----------|\n");

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

printf("\t\t|\t 0. 退出系统\t\t\t|\n");

printf("\t\t|\t 1. 输入记录\t\t\t|\n");

printf("\t\t|\t 2. 查找记录\t\t\t|\n");

printf("\t\t|\t 3. 删除记录\t\t\t|\n");

printf("\t\t|\t 4. 修改记录\t\t\t|\n");

printf("\t\t|\t 5. 插入记录\t\t\t|\n");

printf("\t\t|\t 6. 记录排序\t\t\t|\n");

printf("\t\t|\t 7. 记录个数\t\t\t|\n");

printf("\t\t|\t 8. 显示记录\t\t\t|\n");

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

printf("\t\t请输入您的操作(0-8):");

}

### 12.5.2 录入记录模块

在学生信息管理系统中录入学生信息模块主要用于根据提示信息将学生学号、姓名、语文、数学、英语等科目成绩依次输入，录入结束后系统会自动将学生信息保存到磁盘文件中，并计算出学生的平均分。程序应该具备学生学号的唯一化处理功能。在录入新记录之前，先显示当前文件中已经存在的所有记录，所以调用了show( )函数。

当用户在功能选择界面中按照提示输人数字“1”，即可进入到录入学生信息状态。当磁盘文件有存储记录时，可向文件中追加新的学生成绩信息，运行效果如图17-3 所示。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*录入学生信息\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void input()

{

int i,m=0; /\*变量m表示记录的条数\*/

char ch[2];

FILE \*fp; /\*定义文件指针\*/

if((fp=fopen("stuData.txt","a+"))==NULL) /\*打开指定文件\*/

{

printf("打开文件失败\n");

return;

}

while(!feof(fp))

{

if(fread(&stu[m] ,LEN,1,fp)==1)

m++; /\*统计当前记录条数\*/

}

fclose(fp);

if(m==0)

printf("没有数据!\n");

else

{

system("cls");

show(); /\*调用show函数，显示原有信息\*/

}

if((fp=fopen("stuData.txt","wb"))==NULL)

{

printf("打开文件失败\n");

return;

}

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

{

fwrite(&stu[i] ,LEN,1,fp); /\*向指定的磁盘文件写入信息\*/

}

printf("请确认(y/n):");

scanf("%s",ch);

while(strcmp(ch,"Y")==0||strcmp(ch,"y")==0) /\*判断是否要录入新信息\*/

{

printf("number:");

scanf("%d",&stu[m].num); /\*输入学生学号\*/

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

if(stu[i].num==stu[m].num)

{

printf("此学号已存在！");

getch();

fclose(fp);

return;

}

/\*如果学号可用，则继续输入其它数据\*/

printf("name:");

scanf("%s",stu[m].name); /\*输入学生姓名\*/

printf("chinese:");

scanf("%f",&stu[m].chinese); /\*输入语文课成绩\*/

printf("math:");

scanf("%f",&stu[m].math); /\*输入数学课成绩\*/

printf("english:");

scanf("%f",&stu[m].english); /\*输入英语课成绩\*/

stu[m].sum=stu[m].chinese+stu[m].math+stu[m].english; /\*计算出总成绩\*/

if(fwrite(&stu[m],LEN,1,fp)!=1) /\*将新录入的信息写入指定的磁盘文件\*/

{

printf("保存失败!");

getch();

}

else

{

printf("%s saved!\n",stu[m].name);

m++;

}

printf("是否继续?(y/n):"); /\*询问是否继续\*/

scanf("%s",ch);

}

fclose(fp);

printf("操作成功!\n");

}

### 12.5.3 查找记录模块

查找记录模块的主要功能是根据用户输入的学生学号对学生数据进行搜索，在功能菜单中与之对应的输入是2，如果查找成功，则显示此学生的数据，运行结果如图12- 所示。

如果所查的学生学号在文件中不存在，则系统会给出“查找失败!”的提示，运行结果如图12- 所示。

如果文件在打开时便没有记录，在进行查询时，会先显示“文件无记录! ”的提示，运行结果如图12- 所示。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*自定义查找函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void search()

{

FILE \*fp;

int snum,i,m=0;

char ch[2];

if((fp=fopen("stuData.txt","rb"))==NULL)

{

printf("打开文件失败\n");

return;

}

while(!feof(fp))

if(fread(&stu[m],LEN,1,fp)==1)

m++;

fclose(fp);

if(m==0)

{

printf("文件无记录!\n");

return;

}

printf("请输入需要删除的学生学号:");

scanf("%d",&snum);

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

if(snum==stu[i].num) /\*查找输入的学号是否在记录中\*/

{

printf("学生数据已找到，确认显示?(y/n)");

scanf("%s",ch);

if(strcmp(ch,"Y")==0||strcmp(ch,"y")==0)

{

printf("学号 姓名 chinese math english sum\t\n");

printf(FORMAT,DATA); /\*将查找出的结果按指定格式输出\*/

break;

}

}

if(i==m)

printf("查找失败!\n"); /\*未找到要查找的信息\*/

}

### 12.5.4 删除记录模块

删除学生信息模块主要的功能是从磁盘文件中将学生的信息读出来，从读出的信息中将要删除的学生记录找到，然后将此学生的记录数据删除。删除的过程中，后续记录会依次前移并覆盖被删除的记录。删除记录功能模块对应的菜单选项为3，调用删除记录函数del( )后，运行结果如图12- 所示。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*自定义删除函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void del()

{

FILE \*fp;

int snum,i,j,m=0;

char ch[2];

if((fp=fopen("stuData.txt","r+"))==NULL)

{

printf("打开文件失败\n");return;}

while( !feof(fp) )

if(fread(&stu[m],LEN,1,fp)==1)

m++;

fclose(fp);

if(m==0)

{

printf("文件无记录!\n");

return;

}

show();

printf("请输入需要删除的学生学号:");

scanf("%d",&snum);

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

{

if(snum==stu[i].num)

{

printf("数据已找到，是否确认删除?(y/n)");

scanf("%s",ch);

if(strcmp(ch,"Y")==0||strcmp(ch,"y")==0) /\*判断是否要进行删除\*/

for(j=i;j<m;j++)

stu[j]=stu[j+1]; /\*将记录依次移到前面的位置\*/

m--; /\*记录的总个数减1\*/

if((fp=fopen("stuData.txt","wb"))==NULL)

{

printf("打开文件失败\n");

return;

}

for(j=0;j<m;j++)/\*将更改后的记录重新写入指定的磁盘文件中\*/

if(fwrite(&stu[j] ,LEN,1,fp)!=1)

{

printf("对不起，保存失败!\n");

getch();

}

fclose(fp);

printf("删除成功!\n");

return;

}

}

printf("没有找到要删除的信息！\n");

}

### 12.5.4 修改记录模块

修改记录模块函数modify( )函数对应的功能菜单中的数字4，函数会先根据用户输入的学号查找，查找成功后，列出此学生的所有信息。然后给出提示，引导用户一步一步的对学生信息进行修改，如图12-所示。如果不存在记录，则会提示“没有找到匹配信息！”。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*自定义修改函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void modify()

{

FILE \*fp;

int i,j,m=0,snum;

if((fp=fopen("stuData.txt","r+"))==NULL)

{

printf("打开文件失败\n");

return;

}

while(!feof(fp))

if(fread(&stu[m],LEN,1,fp)==1)

m++;

if(m==0)

{

printf("文件无记录!\n");

fclose(fp);

return;

}

show();

printf("请输入你要修改的学生的学号\n");

printf("学号是:");

scanf("%d",&snum);

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

{

if(snum==stu[i].num)/\*检索记录中是否有要修改的信息\*/

{

printf("学生数据已找到，请修改!\n");

printf("name:");

scanf("%s",stu[i].name);/\*输入名字\*/

printf("chinese:");

scanf("%f",&stu[i].chinese);/\*输入语文课成绩\*/

printf("math:");

scanf("%f",&stu[i].math);/\*输入数学课成绩\*/

printf("english:");

scanf("%f",&stu[i].english);/\*输入英语课成绩\*/

printf("modify successful!");

stu[i].sum=stu[i].chinese+stu[i].math+stu[i].english;

if((fp=fopen("stuData.txt","wb"))==NULL)

{

printf("打开文件失败\n");

return;

}

for(j=0;j<m;j++)/\*将新修改的信息写入指定的磁盘文件中\*/

if(fwrite(&stu[j] ,LEN,1,fp)!=1)

{

printf("保存失败!");

getch();

}

fclose(fp);

return ;

}

}

printf("没有找到匹配信息！\n");

}

### 12.5.5 插入记录模块

插入记录模块的主要功能是在需要的位置插入新的学生记录，在菜单中对应的功能键为数字5，插入记录函数insert( )函数运行后，会先提示记录的插入位置，即需要插入在哪个学生信息的后面，然后程序会进行查找，查找成功后，将后续的所有学生记录依次后移一位，为即将插入的学生记录空出一位置。最后将数据重新写回磁盘。运行结果如图12- 所示。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*自定义插入函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void insert()

{

FILE \*fp;

int i,j,k,m=0,snum;

if((fp=fopen("stuData.txt","r+"))==NULL)

{

printf("打开文件失败\n");

return;

}

while(!feof(fp))

if(fread(&stu[m],LEN,1,fp)==1) m++;

if(m==0)

{

printf("no record!\n");

fclose(fp);

return;

}

printf("请输入数据插入的位置：\n");

scanf("%d",&snum);/\*输入要插入的位置\*/

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

if(snum==stu[i].num)

break;

for(j=m-1;j>i;j--)

stu[j+1]=stu[j];/\*从最后一条记录开始均向后移一位\*/

printf("请输入待插入数据信息.\n");

printf("number:");

scanf("%d",&stu[i+1].num);

for(k=0;k<m;k++)

if(stu[k].num==stu[m].num)

{

printf("此学号已存在！");

getch();

fclose(fp);

return;

}

printf("name:");

scanf("%s",stu[i+1].name);

printf("chinese:");

scanf("%f",&stu[i+1].chinese);

printf("math:");

scanf("%f",&stu[i+1].math);

printf("english:");

scanf("%f",&stu[i+1].english);

stu[i+1].sum=stu[i+1].chinese+stu[i+1].math+stu[i+1].english;

if((fp=fopen("stuData.txt","wb"))==NULL)

{

printf("打开文件失败\n");

return;

}

for(k=0;k<=m;k++)

if(fwrite(&stu[k] ,LEN,1,fp)!=1)/\*将修改后的记录写入磁盘文件中\*/

{

printf("保存失败!");

getch();

}

fclose(fp);

}

### 12.5.6 记录排序模块

记录排序模块实现在功能是对学生总成绩进行从高到低排序，并将排序后的数据写回磁盘中保存，排序函数order( )在功能菜单中对应的数字键是6，排序成功能的显示效果如图12- 所示。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*自定义排序函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void order()

{

FILE \*fp;

struct student t;

int i=0,j=0,m=0;

if((fp=fopen("stuData.txt","r+"))==NULL)

{

printf("打开文件失败!\n");

return;

}

while(!feof(fp))

if(fread(&stu[m] ,LEN,1,fp)==1)

m++;

fclose(fp);

if(m==0)

{

printf("无记录!\n");

return;

}

if((fp=fopen("stuData.txt","wb"))==NULL)

{

printf("打开文件失败\n");

return;

}

for(i=0;i<m-1;i++)

for(j=i+1;j<m;j++)/\*双重循环实现成绩比较并交换\*/

if(stu[i].sum<stu[j].sum)

{

t=stu[i];stu[i]=stu[j];

stu[j]=t;

}

if((fp=fopen("stuData.txt","wb"))==NULL)

{

printf("打开文件失败\n");

return;

}

for(i=0;i<m;i++) /\*将重新排好序的内容重新写入指定的磁盘文件中\*/

if(fwrite(&stu[i] ,LEN,1,fp)!=1)

{

printf("%s 保存失败!\n");

getch();

}

fclose(fp);

printf("操作成功\n");

}

### 12.5.7 记录个数显示模块

如果想查看当前文件中学生记录的总数，可用total( )函数实现，对应的功能菜单的数字键7，运行结果如图12- 所示。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*显示学生总人数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void total()

{

FILE \*fp;

int m=0;

if((fp=fopen("stuData.txt","r+"))==NULL)

{

printf("打开文件失败\n");

return;

}

while(!feof(fp))

if(fread(&stu[m],LEN,1,fp)==1)

m++;/\*统计记录个数即学生个数\*/

if(m==0)

{

printf("文件无记录!\n");

fclose(fp);

return;

}

printf("学生总数为 %d 位!\n",m);/\*将统计的个数输出\*/

fclose(fp);

}

### 12.5.7 记录显示模块

记录显示模块show( )可以使文件中的记录按指定的格式打印出来，它对应的功能菜单数字键为8，运行结果如图12-所示。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*显示学生信息\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void show()

{

FILE \*fp;

int i,m=0;

fp=fopen("stuData.txt","rb");

while(!feof(fp))

{

if(fread(&stu[m] ,LEN,1,fp)==1)

m++;

}

fclose(fp);

printf("学 号 姓 名 语 文 数 学 英 语 总 分\t\n");

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

{

printf(FORMAT,DATA); /\*将学生信息按指定格式打印\*/

}

}