



# 实 训 报 告

实践环节 : 程序设计方法实训（上）

项目名称 : 学生成绩管理系统的设计与实现

院（系） : 软件学院

专业班级 : 软件工程 17-03

学 号 : 541713460329

学生姓名 : 苏一恒

指导教师 : 徐洁 王博

学年学期 : 2017-2018 学年第 1 学期

成 绩 : \_\_\_\_\_

# 郑州轻工业学院软件学院

## 实训任务书

### 一、实训名称

实践环节 程序设计方法实训（上）

项目名称 学生成绩管理系统的设计与实现

### 二、学生信息

专业班级 软件工程 17-03 学号 541713460329 姓名 苏一恒

### 三、用户需求及约束条件

1. 用 C 语言实现“学生成绩管理系统”。
2. 设计“学生成绩管理系统”的用户界面，系统启动进入后，只能通过界面菜单指令才能退出。

3. “学生成绩管理系统”具有以下功能：

（1）系统启动后，弹出一个用户界面，能实现人机交互的功能。而且只能通过界面菜单命令实现各种功能（包括系统退出功能），可以设计欢迎词。

（2）**成绩录入**：录入信息包括班级、姓名、学号、五门课程成绩。

（3）**计算平均成绩**：计算每门课程的平均成绩和学生五门课程的平均成绩并存档。

下面的功能必须是第（3）项功能完成以后才能进行。

（4）**成绩输出**：输出学生的信息和对应的原始成绩以及平均成绩；

（5）**成绩排序（按平均成绩）**，输出排序后的学生信息以及对应的课程成绩和平均成绩。

（6）**成绩查找**：两种查找方法，一是按学号查找，二是按姓名查找。

(7) **成绩修改**：查找后修改成绩，修改后重新排序，并输出学生信息和对应课程成绩和平均成绩。

(8) **成绩追加**：增补遗漏的学生以及对应的信息和对应的课程成绩，并计算出平均成绩，增补以后重新排序，并输出新增学生信息和对应课程成绩和平均成绩。

**附加功能：**

(1) **删除成绩**：删除不需要的学生对应的记录（包括姓名、学号、成绩等）。

(2) **成绩分段**：将学生成绩分段。

(3) **文件存储**：把学生信息和对应的课程成绩和平均成绩存入到文件中。

(4) **文件输出**：输出文件中的内容。

(5) **文件导入**：从文件中导入数据到成绩管理系统中，然后进行管理。

#### **四、设计要求、技术参数及设计规格**

1. 原始成绩由老师统一给出，在 Excel 表中。
2. 设计的系统能够运行，提交源程序、目标文件和可执行文件。
3. 提交实训报告，实训报告中列写：1) 软件需求分析；2) 软件规格说明和设计要求；3) 对数据的管理方案；4) “学生成绩管理系统”设计的思路 and 整体框架；5) 各功能或者模块实现的思路（或者用带注释的程序代替）并交待模块之间的接口关系；6) “学生成绩管理系统”设计的结果（可以用一些屏幕拷贝的图来补充说明）。
4. 程序设计环境为 Visual C++ 6.0 或 DEV C++。
5. 所有的设计工作需要同学自己真实地、独立地完成。

#### **五、工作量**

| 天/<br>日期 | 任务描述   | 通过标准  |
|----------|--|---|
| 1        | 上午：查询相关文献，熟悉系统功能。<br>完成三项任务（用文档记录）：1) 软件需求分析；2) 软件规格说明和设计要求；3) 对数据的管理方案。<br>下午：构思“学生成绩管理系统”的思路和整体框架。<br>设计数据结构（命名），设计函数框架（命名，考虑参数的类型和个数，即考虑数据之间的传递关系）。<br>设计被系统调用的 main 函数；实现界面函数；设计各功能子函数的框架； | 文档通过<br>方案通过<br>数据结构通过<br>设计的函数框架通过<br>要求数据传递接口正常，main 函数能运行。 |
| 2        | 上午：实现成绩录入功能；<br>下午：实现计算平均成绩功能；<br>实现成绩输出功能；  | 各子函数调试通过，能被 main 函数成功调用，其子功能能够实现。                             |
| 3        | 上午：实现成绩排序功能；<br>下午：实现成绩查找（按学号查找、按姓名查找），成绩修改功能；   | 各子函数调试通过，能被 main 函数成功调用，其子功能能够实现。                             |
| 4        | 上午：实现追加功能；<br>下午：自我创新，例如图形用户界面，登录功能等；<br>自选附加功能实现；   | 各子函数调试通过。   |
| 5        | 上午：系统完善；撰写实训报告。<br>下午：撰写实训报告<br>程序验收。  | 系统能正确运行，界面友好；各功能能够实现，性能良好，操作方便。                               |
| 备注：      | 特色点自创<br>附加功能自选  |   |

## 六、考核形式、考核时间、考核指标及成绩评定

实训成绩由四部分构成：课堂表现占 10%；实训结果占 30%；实训报告占 30%；

实践操作能力占 30%。

考核形式：1) 参与课堂情况考核；2) 项目完成情况验收；3) 实训报告撰写及成果提交情况。

完 成 期 限： 2017 年 12 月 29 日

指导教师签章： 徐洁 王博

专业负责人签章： 梁树军

教学院长签章： 邓璐娟

2017 年 12 月 16 日

## 实训进度日历

| 实践环节 |           | 程序设计方法实训（上）  |  |                        |            |  |
|------|-----------|--|--|------------------------|------------|--|
| 项目名称 |           | 学生成绩管理系统的设计与实现   |  |                        |            |  |
| 学年学期 |           | 2017-2018 学年<br>第 1 学期   | 专业<br>班级   | 软件工程<br>17-01/02/03/04 | 指导教师<br>姓名 | 孙玉胜、王<br>昌海、马军<br>霞、桑永<br>宣、徐洁、<br>王博、李<br>辉、刘放美 |
| 天    | 上午/<br>下午 | 实训任务详细描述   | 教师实训任务的指导（解决方案、实现步骤、技术路线、难点提示）   |                        |            | 通过标准   |
| 1    | 上午        | 完成：1) 软件需求分析；2) 软件规格说明和设计要求；3) 对数据的管理方案。   | 对软件功能进行分类。<br>用户界面操作简单、功能完备，系统启动后先录入数据，再求平均成绩；平均成绩求出后，其他的功能才能实现。<br>数据结构可以采用数组也可以采用链表。                                     |                        |            | 文档通过验收   |
|      | 下午        | 构思“学生成绩管理系统”的思路和整体框架。<br>设计数据结构（命名），设计函数框架（命名，考虑参数的类型和个数，即考虑数据之间的传递关系）<br>设计被系统调用的 main 函数；设计界面函数； | 如果采用数组管理数据结构，则数组必须按地址传递，这样数据才能<br>在各个函数间共享。<br><br>难点 1：系统的退出，exit(0)函数需要包含 stdlib.h<br>难点 2：清屏函数 clrscr( )；<br>难点 3：数据的传递 |                        |            | 设计的函数框架通过调试，要求数据传递接口正常，main 函数能运行。               |
| 2    | 上午        | 设计数据录入函数，从键盘读数据到数组中，或者创建链表；  | 难点 1：数据输入完后，怎样控制结束？用一个什么标志。<br>难点 2：结构体数组中的子域又是一个数组时，数据不能直接从键盘读入，需用一个临时变量中转一下。<br>难点 3：从键盘录入数据时，要给完全的提示信息，便于操作指导。          |                        |            | 各子函数调试通过，能被 main 函数成功调用                          |
|      | 下午        | 计算平均成绩的函数实现；<br><br>输出原始成绩；  | 难点 4：想一想：在录入数据时数据保存到一数组，那么在算平均值时，怎样能接触到此数组中的数据。<br><br>输出按照一定的格式，美观。   |                        |            | 各子函数调试通过，能被 main 函数成功调用。                         |

|   |    |                  |  |  |
|---|----|------------------|--|--|
| 3 | 上午 | 排序功能实现并输出排序后的成绩。 | 按平均成绩排序。<br>排序的算法可以是冒泡法或者选择法等<br>思考 1：你还可以分别在每个班中按平均成绩排序吗？<br>思考 2：能输出有不及格课程的同学信息和成绩吗？ | 各子函数调试通过，能被 main 函数成功调用。   |
|   | 下午 | 实现查找和修改功能        | 查找到某一学生（按姓名查找，按学号查找），修改某一属性，如果需要重新排序，重新排序。<br>如果要用对半查找法，需要数据是有序的。                      | 各子函数调试通过，能被 main 函数成功调用。   |
| 4 | 上午 | 追加学生的信息；         | 增加后重新排序，重新输出。<br>难点：增加后重新排序  |  |
|   | 下午 | 自定义和附加功能实现       | 例如图形用户界面，登录功能；密码输入显示*；   | 系统能正确运行，界面友好；<br>各功能能够实现，性能良好操作方便。   |
| 5 | 上午 | 完成实训报告           | 难点：清晰、明确、简单、评价合理、有自己的见解、创新。  | 实训报告形式规范，内容完整。<br>实训报告装订整齐。  |
|   | 下午 | 验收程序讨论           |  | 1) 系统能正确运行，界面友好；<br>2) 各功能能够实现，性能良好操作方便；<br>3) 清晰、明确、简洁、中肯进行汇报和问题陈述；<br>4) 按要求上传/上交成果。 |

制订教师： 孙玉胜、王昌海、马军霞、桑永宣、徐洁、王博、李辉、刘放美

# 目 录

|     |              |   |
|-----|--------------|---|
| 1   | 系统分析 .....   | 1 |
| 1.1 | 功能需求 .....   | 1 |
| 1.2 | 非功能性需求 ..... | 1 |
| 2   | 系统设计 .....   | 1 |
| 2.1 | 体系结构设计 ..... | 1 |
| 2.2 | 数据设计 .....   | 1 |
| 2.3 | 模块设计 .....   | 2 |
| 3   | 系统实现 .....   | 3 |
| 3.1 | 欢迎模块 .....   | 3 |
| 3.2 | 密码模块 .....   | 3 |
| 3.3 | 菜单模块 .....   | 3 |
| 3.4 | 录入模块 .....   | 4 |
| 3.5 | 输出模块 .....   | 5 |
| 3.6 | 查找模块 .....   | 5 |
| 3.7 | 删除模块 .....   | 7 |
| 3.8 | 分段模块 .....   | 8 |
| 3.9 | 文件操作模块 ..... | 8 |



|       |                    |    |
|-------|--------------------|----|
| 3.9.1 | 保存到文件中 .....       | 8  |
| 3.9.2 | 展示文件的内容 .....      | 9  |
| 3.9.3 | 将文件的内容导入到软件中 ..... | 10 |
| 3.10  | 结束系统模块 .....       | 10 |
| 4     | 实训感想 .....         | 12 |
| 5     | 附录 .....           | 14 |

# 1 系统分析

此学生成绩管理系统是为管理学生成绩数据的人员提供，为管理学生成绩数据的相关人员提供充足的信息和快捷的查询手段。该系统的编写目的是提高学生成绩管理的工作效率，使人们从乏味的数据登记和统计工作中解脱出来，保证工作的准确率。

## 1.1 功能需求

在系统中容易添加学生成绩数据的信息，删除学生数据的信息，对于已添加的信息能够进行降序排序，分段，查找，修改等等。另外学生的信息要能够保存到固定的文件中并能够进行下一次的读取。

## 1.2 非功能性需求

### (1)产品质量属性需求

正确性：不允许出现业务错误。

健壮性：在系统出现故障时，系统必须提供相应的数据保护措施，不能丢失数据，并提供从新恢复正常工作的机制。

可靠性：除电源硬件、操作系统外不允许异常退出或崩溃。

性能、效率：系统处理业务时间最迟时间小于等于 1 秒。

易用性：使用简单，操作方便。

安全性：用户只能通过系统操作后台数据，不能够对数据库进行直接操作。

可扩展性：可在当前需求基础之上进行功能上的扩展。

兼容性：能够运行在多台主机上。

可移植性：可以够运行在 WindowsXP 以上 Windows 系列操作系统平台上。

### (2)用户界面

①采用类似与 DOS 的界面，用户界面友好。

- ②界面具有一致性，界面规范遵循 DOS 软件界面的规范。
- ③提供错误信息处理。
- ④提供信息提示，提示当前用户的误操作信息和指示信息。
- ⑤提供方便的退出系统的方法
- ⑥用户对程序的维护，最好要有备份

## 2 系统设计

此系统是的设计采用模块化的设计方法，根据使用者的选择逐步调用。采用模块化的程序设计方法，即将较大的任务按照一定的原则分为一个个较小的任务，然后分别设计各个小任务。需要注意的是划分出来的模块应该相对独立但又相关，且容易理解。可以用模块化层次结构图（即模块图）来分析其任务的划分，一般从上到下进行，最上面一层是主模块，下面各层是其上一层模块的逐步细化描述。

### 2.1 体系结构设计

创建结构体数组来储存学生的信息，创建一个 switch 函语句创建一个整体框架，case 下包括各个主干功能，某些主干功能也用 switch 来包括分支功能，程序的各个功能都用函数表示。

### 2.2 数据设计

定义全局结构体方便调用，MAX\_STU 是最大学生数量，方便后期开通增值服务。

```
struct Student          //定义一个学生结构体
{
    char names[30];      //姓名
    long long stu_id;    //学号
    int stu_class;       //班级
    double lesson1;      //高等数学·上
    double lesson2;      //英语视听说
    double lesson3;      //大学计算机基础
    double lesson4;      //软件工程导论
    double lesson5;      //C 语言程序设计
    double avg;
}student[MAX_STU];

char password_real[30] = "123456";    //在此处设置密码
int now_index=0;                       //开始录入第 now_index 个学生
int i;                                  //定义整形 i 用来 for 循环
```

```
double avg1=0,avg2=0,avg3=0,avg4=0,avg5=0,avg0; //为求平均做准备
FILE * fp;
char temp_cls;                                //清屏函数的临时回车选项
```

## 2.3 模块设计

具体模块设计如图 2-1 所示。

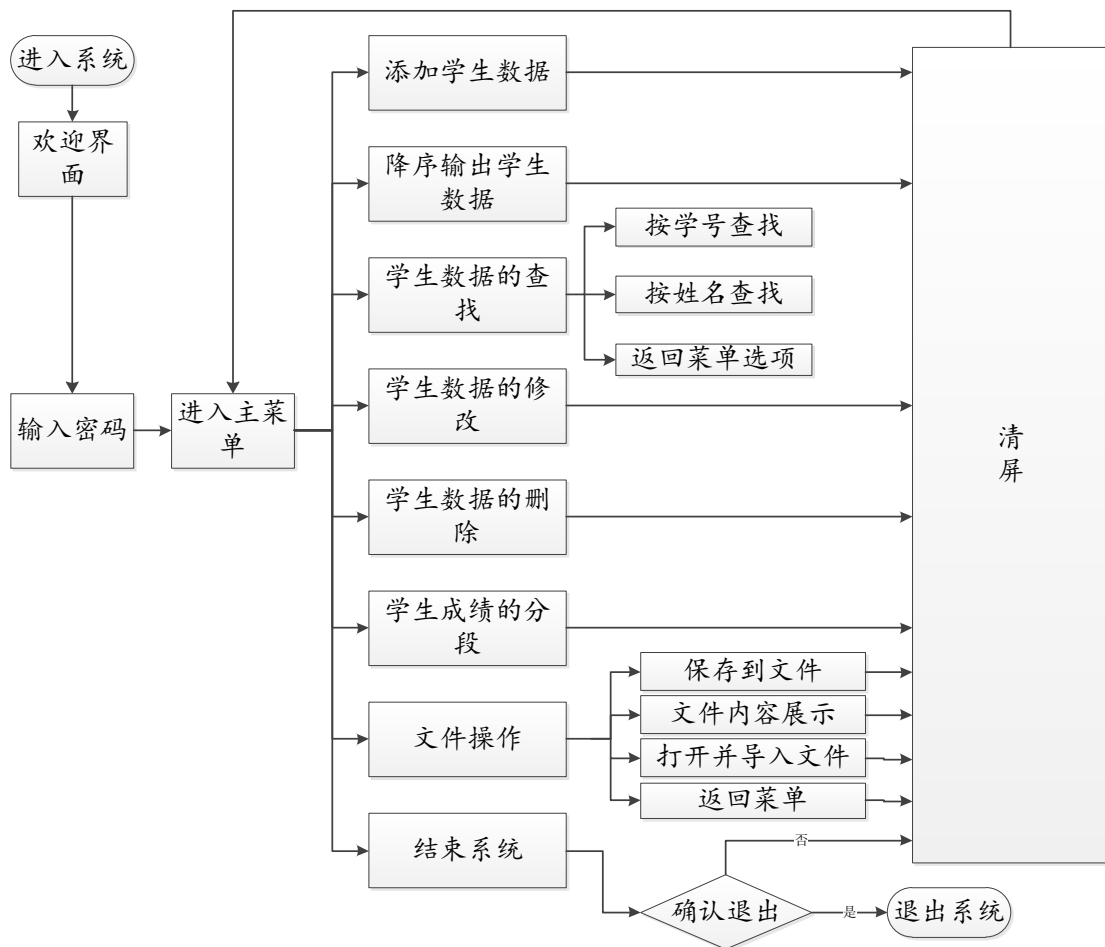


图 2-1 模块设计分析图

## 3 系统实现

模块化的系统实现，参考上述模块设计图。

### 3.1 欢迎模块

介绍公司和慰问用户，使用户能够感受到软件制作团队的认真与用心。在欢迎模块中附加温馨提示，以防给使用者带来不舒适的软件市容体验，另外附加上软件的升级会员版的条件的途径。欢迎模块如图 3-1 所示。



图 3-1 欢迎模块实现图

### 3.2 密码模块

在此程序的欢迎界面之后紧接着是输入密码的界面，输入密码要求输入正确的密码。输入密码的实现是将用户输入的一串字符和系统内本身定义好的字符串进行比较，用 `strcmp()` 验证时候相等，如果判断正确，通过进入系统，如果不成立，则进入循环输入状态，直到输入 5 次为止。累计输入 5 次错误则退出系统。(密码是 123456)

### 3.3 菜单模块

菜单模块的实现主要包括两部分：

第一部分是通过 `printf()` 函数将菜单选项提供给用户进行选择，根据用户接下来的操作来判定下一步的操作。

第二部分是根据用户的选项调用已经封装好的各个功能的函数，代码如下。本功能的实现是通过 `switch()` 函数来找到与用户输入的功能序号相对应的封装

函数。利用 goto 语句进行抛出异常。其中 cls();是清屏函数，它的实现是调用 windows.h 头文件下的 system();来实现。

```
loop_menu:scanf("%d",&menu_id);
    switch(menu_id)                //选择菜单
    {
        case 1: stu_add();break;    //添加
        case 2: stu_output();cls();break; //输出
        case 3: stu_find();cls();break; //查找
        case 4: stu_revise();cls();break; //修改
        case 5: stu_delete();cls();break; //删除
        case 6: stu_subsection();cls();break; //分段
        case 7: stu_fileClass();break; //文件操作
        case 8: stu_over();cls();break; //结束
        default : {
                                fflush(stdin);
                                printf("您的输入有误，请重新输入：");
                                goto loop_menu; //循环该菜单
                            }
    }
```

菜单模块实现图如图 3-2 所示。



```
密码输入正确，欢迎使用！
*****
1、添加新的学生数据          2、降序输出当前学生数据(包括平均成绩)
3、学生数据的查找            4、学生数据的修改
5、学生数据的删除            6、学生成绩的分段
7、文件操作                  8、结束系统
*****
请输入你要进行的操作前面的序号: _
```

图 3-2 菜单模块实现图

## 3.4 录入模块

录入模块的实现是让用户通过使用空格分隔来录入每一次代码输入的各个不同的值，然后赋予给结构体成员的不同变量。每一次输入之后，提示用户是否继续输入，如果继续输入那么就重复上述录入的步骤，以便于能够实现多次录入的操作。

如果用户输入的内容不符合规定，则给出提示后加入清楚缓冲区的方法 fflush(stdin);然后重新录入，直用户输入完毕后使用 N\n 来结束录入。在本模块中保留输入有误的提示语，并循环到上一步，这样就能够很容易地使系统具有充足的容错性。

录入模块实现图如图 3-3 所示。

```
*****
1、添加新的学生数据          2、降序输出当前学生数据(包括平均成绩)
3、学生数据的查找          4、学生数据的修改
5、学生数据的删除          6、学生成绩的分段
7、文件操作                8、结束系统
*****
请输入你要进行的操作前面的序号:1
请依次输入学生的姓名、学号、班级(阿拉伯数字)、以及高等数学·上、英语听说、大学计算机基础、软件工程导论、C语言程序设计的成绩，并以空格分隔
小红 123 4 56 74 85 95 65
```

图 3-3 录入模块实现图

### 3.5 输出模块

输出模块首先通过 if 语句判断是否有学生数据的情况，当全局变量 now\_index 数值恒等于 0 时代表学生数据为 0，所以提示用户不能进行此操作。紧接着回到菜单。如果判断当前软件录入人数大于 0，则跳过 if 语句的 return 0 操作。此时紧接着调用求平均函数 avg\_student();来计算每个学生的平均成绩，然后调用排序函数，将学生的成绩降序排列。降序的实现是通过冒泡排序法对全局结构体成员进行降序排列，然后在输出模块的下一部分 for 循环顺序输出，依次实现降序输出学生成绩的目的。在最后调用 avg\_lesson();函数来求取没课成绩的平均成绩并在输出模块的下方输出。avg\_lesson();的实现是定义全局变量总成绩，将之与当前 now\_index 相除求取平均值并通过.2lf 保留两位小数。输出模块实现图如图 3-4 所示。

```
*****
1、添加新的学生数据          2、降序输出当前学生数据(包括平均成绩)
3、学生数据的查找          4、学生数据的修改
5、学生数据的删除          6、学生成绩的分段
7、文件操作                8、结束系统
*****
请输入你要进行的操作前面的序号:2
-----
姓名      学号      班级      高等数学·上      英语听说      大学计算机基础      软件工程导论      C语言程序设计      平均成绩
张三      124      3        65.00           98.00         78.00           45.00           98.00           76.80
李明      123      2        56.00           89.00         78.00           45.00           65.00           66.60
李四      125      3        65.00           98.00         98.00           12.00           0.00           54.60
小红      126      1        45.00           78.00         20.00           20.00           12.00           35.00
-----
每门课程的平均成绩:      57.75      90.75      68.50      30.50      43.75      58.25
-----
重新选择菜单选项请回车后操作!
```

图 3-4 输出模块实现图

### 3.6 查找模块

查找模块的实现的关键在于查找到对应学生的下标，通过找到下标来找到对应的学生结构体成员，并输出。查找模块首先通过 if 语句判断是否有学生数据的情况，当全局变量 now\_index 数值恒等于 0 时代表学生数据为 0，所以提示用户不能进行此操作。紧接着回到菜单。如果判断当前软件录入人数大于 0，则跳过



if 语句的 return 0 操作。代码进行下一步：查找之前首先查找模块分两种情况进行查找：

```
loop_find:
scanf("%d",&find_index);
switch(find_index)
{
    case 1:find_id();break;           //通过学号查找
    case 2:find_name();break;        //通过姓名查找
    case 3:return 0;
    default:                          //假设查找失败
    {
        fflush(stdin);
        printf("您的输入有误请重新输入:");
        goto loop_find;
    }
}
```

按学号查找 find\_id();通过输入的的学生序号与录入的数据中的学生相吻合时（用 if 条件语句进行判断），找到对应学生标号，根据学生标号找到结构体成员，然后进行下一步。

按姓名查找 fin\_name();将输入的学生姓名字符串和录入的所有学生数据用 strcmp();相比较，但判断通过时提取对应下标到下一步。

找到下标之后，提示使用者输入要修改的成绩编号并输入成绩，通过重新赋值的做法将新值录入，旧值释放。

在该模块中附加进入候选项可以直接返回到菜单中的功能，直接用 return 0;实现。

查找模块实现图如图 3-5 所示。

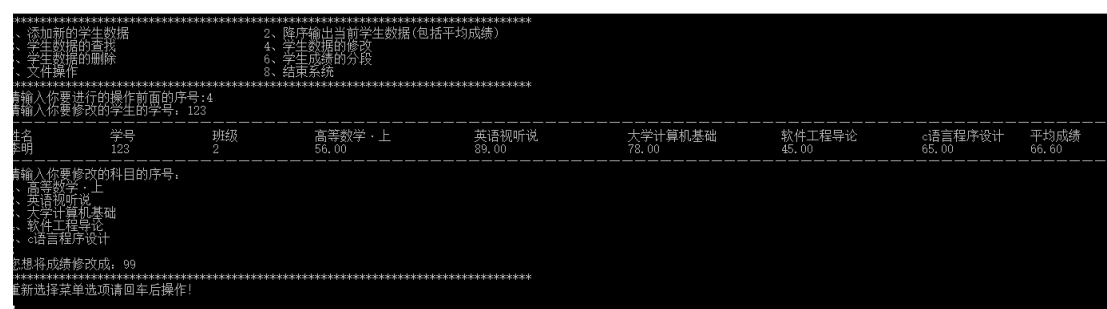


图 3-5 查找模块实现图

### 3.7 删除模块

删除模块的实现和查找模块相似，同样是查找到对应下标，然后将下标对应处成员向前补位。删除模块首先通过 if 语句判断是否有学生数据的情况，当全局变量 now\_index 数值恒等于 0 时代表学生数据为 0，所以提示用户不能进行此操作。紧接着回到菜单。判断通过后，通过将用户输入的学号存入 long long 型变量 revise\_id 中。顺序循环将 revise\_id 与录入的学生的学号对比，如果没找到则抛出异常，如果找到即当恒等于时 break 跳出循环，此时全局变量 i 就是对应下标，传 i 值到 delete\_way(int t); 的参数中，删除的方法是从下标开始的学生结构体成员向前补位，将 now\_index 自减，此时如果使用输出函数就能顺序输出无 student[i] 的结构体数组。

delete\_way(); 的关键方法如下：

```
void delete_way(int t)
{
    int h;
    int temp_loop_num = now_index - t - 1;
    for(h=0; h<MAX_STU-1; h++) //从删除处向前补位
    {
        student[t] = student[t+1];
        t++;
    }
    now_index--;
}
```

删除模块实现图如图 3-6 所示。

```
*****
1、添加新的学生数据          2、降序输出当前学生数据(包括平均成绩)
3、学生数据的查找          4、学生数据的修改
5、学生数据的删除          6、学生成绩的分段
7、文件操作                8、结束系统
*****
请输入你要进行的操作前面的序号:5
请输入你要删除的学生的学号: 125
该学生数据已成功删除!
*****
重新选择菜单选项请回车后操作!
```

图 3-6 删除模块实现图

### 3.8 分段模块

数据的分段，是通过排序后的学生数据顺序输出，与直接输出不同的是，数据的分段还将数据在 5 个分段的学生统计出来并加上美化边框的处理，很容易能让学生找到对应人数。分段模块首先通过 if 语句判断是否有学生数据的情况，当全局变量 `now_index` 数值恒等于 0 时代表学生数据为 0，所以提示用户不能进行此操作。紧接着回到菜单。判断通过后，依次取不同分段的人数，通过循环 5 次不同的次数来输出不同的人并加上对应统计数据 and 边框。通过 5 次循环，展现处用户想要的效果。分段函数实现图如图 3-7 所示。

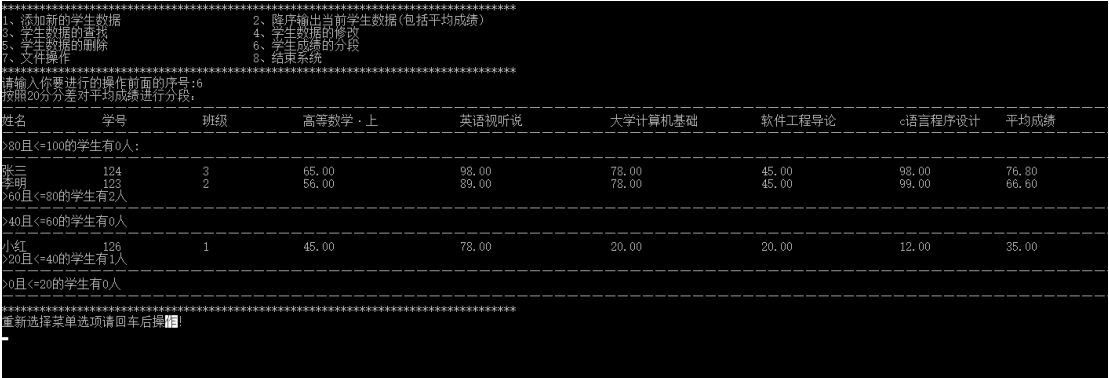


图 3-7 分段模块实现图

### 3.9 文件操作模块

文件操作的模块提供给用户 4 个选项：

- (1)保存到文件中
- (2)文件内容的展示
- (3)打开并导入软件
- (4)返回主菜单

通过 `switch()`将 4 个选型联系起来。

#### 3.9.1 保存到文件中

定义 `char` 型指针 `buffer`，为了输出路径。判断并抛出文件打开的异常。如果文件不存在则自动创建该文件，将标题先输出到 `Student datas.txt` 中，再循环

now\_index 次将各个数据输出到文件中并在 dos 显示屏上显示以验证是否输入正确。最后成功后将提示用户输出到文件成功，并通过 getcwd();显示出路(此函数是在头文件 direct.h 中).然后关闭文件。保存文件的实现图如图 3-8 所示。



图 3-8 保存文件的实现图

3.9.2 展示文件的内容

展示文件的内用的原理是从头到尾以输出字符的方法依次输出到显示屏上。以只读形式打开文件后，核心代码如下

```
char ch;
while((ch=fgetc(fp)) != EOF)          //一直执行到无字符
{
    putchar(ch);
}
fclose(fp);
return 0;
}
```

最后关闭文件，输出完成。输出文件内容的实现图如图 3-9 所示。

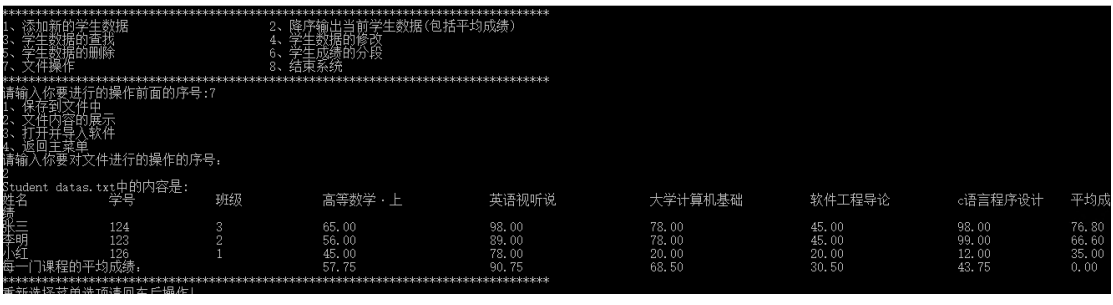


图 3-9 输出文件内容的实现图

### 3.9.3 将文件的内容导入到软件中

文件的导入首先需要找到对应行数 `file_index`。通过下面的代码实现。

```
int line = 0;
for(;;)          //找到共有多少行数据
{
    if((ch = fgetc(fp)) == '\n')
    {
        line++;
    }
    if(ch == EOF)
        break;
}
```

判断对应文件的'\n'找到与行数+2 并-2 存入到局部变量 `file_index` 中。然后  
通过 `rewind(fp);`将文件指针 `fp` 指向文件的开头。

通过 `fscanf(fp,"%[^\\n]",temp_total[0]); fgetc(fp);` 将第一行的标题读取出来  
存入临时变量中并换行

此时判断文件中是否录入数据，当 `line-2` 小于等于 0 时则代表未录入。如果  
判断录入数据，就循环将数据提取出来分别赋予给学生结构体数组中的成员变量。  
导入文件的实现图如图 3-10 所示。



The screenshot shows a C program interface with a menu and a table of student data. The menu includes options for adding, deleting, and displaying student data, as well as file operations. The table displays student information including name, ID, class, and scores for various subjects.

| 姓名 | 学号  | 班级 | 高等数学·上 | 英语视听说 | 大学计算机基础 | 软件工程导论 | C语言程序设计 | 平均成绩  |
|----|-----|----|--------|-------|---------|--------|---------|-------|
| 张三 | 124 | 3  | 65.00  | 98.00 | 78.00   | 45.00  | 98.00   | 76.80 |
| 李明 | 123 | 2  | 56.00  | 89.00 | 78.00   | 45.00  | 99.00   | 66.60 |
| 小红 | 126 | 1  | 45.00  | 78.00 | 20.00   | 20.00  | 12.00   | 35.00 |

图 3-10 导入文件的实现图

### 3.10 结束系统模块

结束模块判断用户是否确认退出程序，提示用户保存数据后再进行退出，也  
可以选择退出程序。此时一定要判断用户是否需要退出系统，如果用户没有保存  
文件就退出系统可能将为下次的操作带来不便。这个 `if` 条件的使用是必须存在  
的。

结束模块的实现图如图 3-11 所示。

```
密码输入正确，欢迎使用！
*****
1、添加新的学生数据          2、降序输出当前学生数据(包括平均成绩)
3、学生数据的查找          4、学生数据的修改
5、学生数据的删除          6、学生成绩的分段
7、文件操作                8、结束系统
*****
请输入你要进行的操作前面的序号:8
温馨提示：您最好确认数据保已经存在文件中或者不需要此操作。
如果需要返回到菜单请输入0
1
离开的时候请带走我真挚的祝福，感谢有你，再见！

-----
Process exited after 5.135 seconds with return value 0
请按任意键继续. . .
```

图 3-11 结束模块的实现图

## 4 实训感想

### (1) 开发经验

通过对本题目的开发，体会到要掌握以下几点内容。

①大程序的设计风格。按“自顶向下，逐步细化，模块化”的方法进行程序设计。

②编写主函数，并进行测试与调试。当被调函数又需要调用其他函数时，也要遵循逐步细化的原则。C 语言提供丰富的库函数，编程序时要善于使用库函数，避免不必要的劳动。定义函数时，应选好参数的个数和数据类型。

③文件使用方法。文件使用完毕后应及时关闭。

### (2) 实训中遇到的问题及解决方法

在这次的实训中，主要遇到的问题是编译环境的不同对程序也不同。该函数未定义的错误中。后来经过老师的指导，终于找到了根源，就是把“clrscr()”改为“system("cls")”就可以了。

### (3) 设计中尚存的不足之处

关于这次的程序，也有挺多不足的地方。首先，我觉得程序还不够完善，有些地方甚至有点乱；其次，有些地方好像功能有点重复了；最后，就是该程序界面不够美观大方。

### (4) 感想和心得体会

通过这次实训，我感触甚深，要把所学的理论知识运用于解决实际问题不仅要付出艰辛的劳动，还得要有科学的方法和坚持不懈的努力。这次实训让我进一步理解了“编程是什么”，“如何去编程”，“编程的目的又是什么”，,,,发现自己的知识竟是如此贫乏，顿时感到了压力的存在，从而也激发了我的学习热情。它将成为我今后不断进取、永不懈怠的动力。

通过本次 C 语言上机实验，我对这个介于人类与非人类之间的计算机编程语言有了一定的体验。其间开心过、郁闷过、无奈过、彷徨过……随着实验的胜利成功与实验报告的胜利完成，有点微微的自豪感使人难忘。感谢王博老师和徐洁老师的指导，感谢实验课上小老师们的耐心指点，也感谢我在

实验中经历过的点点滴滴……伴随着学习的深入，我发现高深的东西还有很多很多，等待着我自己去挖掘。对 C 语言，我会更加努力

在整个实训过程中，我能够坚持虚心的向老师以及同学请教问题，它使我明白了虚心好学的重要性，更让我懂得了如何跟老师同学去相处、去沟通。没有老师和同学们的无私帮助，我不可能圆满完成这次实训任务。感谢帮助过我的所有老师和同学们，要列出所有帮助过我的人的名字是不现实的，因为有些困难是通过因特网解决的，我甚至不知道他们的名字，在此，谨向他们一并表示感谢。



## 5 附录

头文件和相关代码如下

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <direct.h>           //使用返回绝对路径函数 char* getcwd(char* buffer ,int
len);
#include <windows.h>
#define MAX_STU 1000         //最大储存 1000 个学生数据
#define PASSWORD_INPUT_MAX 5 //最大输入的登录密码次数

void Welcome();              //定义一个欢迎界面
void menu();                 //定义一个菜单
void stu_add();              //定义学生数据的添加
int stu_output();            //定义学生数据的输出
int stu_find();              //定义学生数据的查找
int stu_revise();            //定义学生数据的修改
void stu_delete();           //定义学生数据的删除
void delete_way(int t);      //定义删除学生数据的方法
void stu_subsection();       //定义学生数据的分段
void stu_fileClass();        //定义文件的处理
int stu_over();              //定义结束的内容
int file_storage();          //定义文件的储存
int file_show();             //定义文件的输出
int file_import();           //定义文件的导入
void avg_lesson();           //定义求每门课的平均成绩
void avg_student();          //定义每个学生的平均成绩并赋值入结构体数组中
void sort();                 //定义排序函数
void find_id();              //定义按学号查找函数
void find_name();            //定义按姓名查找函数
void cls();                  //定义人性化的清屏函数
void password_input();       //定义密码的输入

struct Student               //定义一个学生结构体
{
    char names[30];          //姓名
    long long stu_id;        //学号
    int stu_class;           //班级
    double lesson1;          //高等数学 • 上
    double lesson2;          //英语视听说
    double lesson3;          //大学计算机基础
```

```

double lesson4;          //软件工程导论
double lesson5;          //C 语言程序设计
double avg;
}student[MAX_STU];

char password_real[30] = "123456";          //在此处设置密码
int now_index=0;          //接下来开始录入第 now_index 个学
生
int i;          //定义整形 i 用来 for 循环
double avg1=0,avg2=0,avg3=0,avg4=0,avg5=0,avg0; //为求平均做准备
FILE * fp;
char temp_cls;          //清屏函数的临时回车选项

```

本次兼任小老师所带的学生有：宋河彭，王永祥，汪健，牛涛杰等