数据结构

课程设计报告

题 目： 学生宿舍管理系统

学 院： 计算机与控制工程学院

专业班级： 软件161

学生姓名： 江雪

指导教师： 王一萍

2018 年12 月26日

数据结构课程设计评分表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 班级 | 软件161 | | 姓名 | 江雪 | 指导教师 | 王一萍 | |
| 题目：学生宿舍管理系统 | | | | | | | |
| 评分标准 | | | | | | | |
| 评分标准 | | 分数权重 | 评分的依据 | | | | 得分 |
| A | | C | |
| 选题 | | 10 | 选题符合大纲要求，题目较新颖，工作量大 | | 选题基本符合大纲要求，工作量适中 | |  |
| 工作态度 | | 10 | 态度端正，能主动认真完成各个环节的工作，不迟到早退，出勤好。 | | 能够完成各环节基本工作，出勤较好。 | |  |
| 存储结构、算法描述 | | 20 | 能正确选择存储结构，定义准确，算法流程图或类C语言描述的算法准确无误 | | 能正确选择存储结构，算法流程图或类C语言描述的算法基本准确 | |  |
| 独立解决问题的能力 | | 10 | 具有独立分析、解决问题能力，有一定的创造性，能够独立完成软件的设计与调试工作，程序结构清晰，逻辑严谨，功能完善。 | | 有一定的分析、解决问题能力。能够在老师指导下完成软件的设计与调试工作，程序功能较完善。 | |  |
| 答辩问题回答 | | 20 | 能准确回答老师提出的问题 | | 能基本准确回答老师提出的问题 | |  |
| 程序运行情况 | | 10 | 程序运行正确、界面清晰，测试数据设计合理。 | | 程序运行正确、界面较清晰，能给出合适的测试数据。 | |  |
| 课程设计报告 | | 20 | 格式规范，层次清晰，设计思想明确，解决问题方法合理，体会深刻。 | | 格式较规范，设计思想基本明确，解决问题方法较合理。 | |  |
| 总分 | | | | | | |  |
| 指导教师（签字）： | | | | | | | |

注：介于A和C之间为B级，低于C为D级和E级。按各项指标打分后，总分在90～100为优，80～89为良，70～79为中，60～69为及格，60分以下不及格。

1 问题描述

学生宿舍管理系统对于一个学校来说是必不可少的组成部分。目前好多学校还停留在宿舍管理人员手工记录数据的最初阶段，手工记录对于规模小的学校来说还勉强可以接受，但对于学生信息量比较庞大，需要记录存档的数据比较多的高校来说，人工记录是下个档麻烦的。而且当查找某条记录时，由于数据量庞大。还只能靠人工取一条条的查找，这样不但麻烦还浪费了许多时间，效率也比较低。。

宿舍管理查询系统，是以学校的管理方式为实例而设计的一种实用型管理查询系统。本系统最大的特点是通用性、简单操作性。随着学校住宿学生的增多，宿舍管理人员的负担越来越重，本系统是为了让所有宿舍管理人员能从繁重的工作中解脱出来，实现无纸化办公，使工作更有条理、更方便、更有效率。

由于本系统是为了方便宿舍管理人员实现宿舍管理查询而开发的，系统本无任何信息数据，要建立数据文件，其必须具有信息录入功能，能通过别的途径导入大量数据，但不能实现信息存盘。信息录入后都保存在内存中并不能直观表现出来，因此需要设计一个信息显示功能，信息的显示应该便于查阅，所以本系统应该具备按各种关键字显示的功能。同时，管理人员可能需要对某个学生进行查找，所以本系统需具有可以按关键字进行排序和查询的功能。由于有些同学因为不同原因而离校，所以本系统需要设计删除功能，而针对有新同学入校，本系统需要具有插入功能，这样便能实现信息的及时修改。当管理人员操作完毕需要退出时，需提供退出选项，便于使用者退出系统。

2 数据结构描述

线性表的[逻辑结构](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91%E7%BB%93%E6%9E%84/9663235" \t "https://baike.baidu.com/item/%E7%BA%BF%E6%80%A7%E8%A1%A8/_blank)简单，在数据结构逻辑层次上细分，便于实现和操作。线性表中链表最明显的好处就是，常规[数组](https://baike.baidu.com/item/%E6%95%B0%E7%BB%84/3794097" \t "https://baike.baidu.com/item/%E9%93%BE%E8%A1%A8/_blank)排列关联项目的方式可能不同于这些数据项目在[记忆体](https://baike.baidu.com/item/%E8%AE%B0%E5%BF%86%E4%BD%93/3029693" \t "https://baike.baidu.com/item/%E9%93%BE%E8%A1%A8/_blank)或[磁盘](https://baike.baidu.com/item/%E7%A3%81%E7%9B%98/2842227" \t "https://baike.baidu.com/item/%E9%93%BE%E8%A1%A8/_blank)上顺序，数据的存取往往要在不同的排列顺序中转换，链表允许插入和移除表上任意位置上的[节点](https://baike.baidu.com/item/%E8%8A%82%E7%82%B9/865052" \t "https://baike.baidu.com/item/%E9%93%BE%E8%A1%A8/_blank)，由于不必须按顺序存储，链表在插入的时候就可以达到O(1)的复杂度，比另一种线性表顺序表快得多，所以本系统采用线性表的链式存储结构。

2.1 逻辑结构和存储结构

（1）线性表的链式存储

链表是一种物理[存储单元](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E5%8D%95%E5%85%83/8727749" \t "https://baike.baidu.com/item/%E9%93%BE%E8%A1%A8/_blank)上非连续、非顺序的[存储结构](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E7%BB%93%E6%9E%84/350782" \t "https://baike.baidu.com/item/%E9%93%BE%E8%A1%A8/_blank)，由一系列结点（链表中每一个元素称为结点）组成，结点可以在运行时动态生成。每个结点包括两个部分：一个是存储[数据元素](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%85%83%E7%B4%A0" \t "https://baike.baidu.com/item/%E9%93%BE%E8%A1%A8/_blank)的数据域，另一个是存储下一个结点地址的[指针](https://baike.baidu.com/item/%E6%8C%87%E9%92%88/2878304" \t "https://baike.baidu.com/item/%E9%93%BE%E8%A1%A8/_blank)域。使用链表结构可以克服[数组](https://baike.baidu.com/item/%E6%95%B0%E7%BB%84/3794097" \t "https://baike.baidu.com/item/%E9%93%BE%E8%A1%A8/_blank)链表需要预先知道数据大小的缺点，链表结构可以充分利用计算机内

存空间，实现灵活的内存动态管理。

typedef struct Student //结构体

{

int data; //数据域

int length; //当前长度

int listsize; //当前分配的存储容量

struct Student \*next;//指针域

}Stu;

（2）ADT

ADT，即抽象数据类型，是一个实现包括储存数据元素的存储结构以及实现基本操作的算法。在这个数据抽象的思想中，数据类型的定义和它的实现是分开的。以下为本系统的ADT。

ADT system{

数据对象{Stu，lh，roomnum，bednum， num，name[20]，xy[100]，bj[15]，sex[10]}

基本操作{

init(l)：初始化函数

void login()：登录界面

create(l)：线性表创建函数

menu()：主菜单函数

input(l)：学生信息添加函数

output(l)：学生信息输出函数

sort (l)：排序函数**//**分别按宿舍楼号、房间号、床位号、学号进行排序（选择排序）

disp()：返回主界面函数

insert(l) ：插入函数

find(l)：查找函数**//**分别按宿舍楼号、房间号、床号、学院、班级、姓名、学号、性别查找进行函数查找

correct(l)：修改函数

delete(l) ：删除函数

write\_file(l)：写入文件函数

read\_file()：读取文件函数

}

} ADT system

（3）结构体数组

本系统采用结构体数组。结构体就是一个可以包含不同数据类型的一个结构，它是一种可以自己定义的数据类型。结构体是由一系列具有相同类型或不同类型的数据构成的数据集合，可以在一个结构中声明不同的数据类型，并且相同结构的结构体变量是可以相互赋值的，而这是数组做不到的，因为数组是单一数据类型的数据集合，它本身不是数据类型（而结构体是），数据名称是常量指针，所以不可以作为左值进行运算，所以数组之间就不能通过数组名称相互复制，即使数据类型和数组大小完全相同。同时，结构体便于分配内存，具有节省空间内存和高效率的优点，针对此学生宿舍信息管理系统储存信息量大、需要便于存取信息的特点，由以上原因故采取如下结构体。

typedef struct Student//结构体

{

int lh; //宿舍楼号

int roomnum; //房间号

int bednum; //床位号

int num; //学号

char name[20]; //姓名

char xy[100]; //学院

char bj[15]; //班级

char sex[10]; //性别

}Stu;

2.2 主要算法

（1）选择排序

链表选择排序是使用链表实现选择排序，一般的选择排序是在数组中实现的，与在数组中实现的选择排序不同的是，链表中选择排序时每次交换数据是通过交换链表的节点来实现的，由于数据是存放与链表的节点中的，所以交换节点就等价于交换了数据的顺序。选择排序算法移动的次数已知，并且每次都是最优解。选择排序与输入无关，即无论输入数组是正序，倒序，还是乱序，其时间复杂度都一样因此本系统采

用选择排序的方式按关键字（宿舍楼号、床位号、房间号以及学号）对数据文件进行排序。

（2）顺序查找

顺序查找算法是按照序列原有顺序对数组进行遍历比较查询的基本查找算法，让关键字与队列中的数从最后一个开始逐个比较，直到找出与给定关键字相同的数为止。顺序查找实现简单，所以本系统采用了顺序查找算法对学生信息数据文件进行查找、插入和删除操作。

3 算法描述

3.1 系统功能模块结构

将本系统的功能进行分解，本系统一共包括九个模块：信息录入模块、信息显示模块、信息排序模块、信息查询模块、信息插入模块、信息删除模块、写入文件模块和读取文件模块。

功能结构图是功能内部的结构图，用来表示复杂功能的内部功能，包含了哪些子功能，将系统的功能进行分解。为了使用户更加清楚本系统的功能有几个部分组成、每部分是什么、以及各有什么作用，将在下面给出本系统的功能结构图如图3-1。

操作流程图可以直观地描述一个工作过程的具体步骤，形象直观，各种操作一目了然，不会产生“歧义性”，对准确了解事情是如何进行的极有帮助，因此，将在下面给出本系的操作流程图如图3-2。

图3-1 功能结构图

图3-2 操作流程图



以下将抽出3个主要功能模块进行详细说明。

3.2信息排序模块

（1）功能

该模块采用选择排序，宿舍管理人员可以根据不同情况的需要，对已录入的数据文件按关键字（楼号、寝室号、床位号、学号）进行排序。

（2）算法描述

选择排序即每一趟在n-i+1（i=1，2，…n-1）个记录中选取关键字最小的记录作为有序序列中第i个记录。再次排序将第i个记录后面的每一个元跟第i个元素进行比较，再持续每次对越来越少的元素重复以上操作，直到没有任何一对数字需要进行比较。

对比数组中前一个元素跟后一个元素的大小，如果后面的元素比前面的元素小则用一个变量k来记住他的位置，接着第二次比较，前面“后一个元素”现变成了“前一个元素”，继续跟他的“后一个元素”进行比较如果后面的元素比他要小则用变量k记住它在数组中的位置(下标)，等到循环结束的时候，我们应该找到了最小的那个数的下标，然后进行判断，如果这个元素的下标不是第一个元素的下标，就让第一个元素跟它交换一下值，这样就找到整个数组中最小的数了。然后找到数组中第二小的数，让其跟数组中第二个元素交换一下值，以此类推。

（3）具体语言实现

在本宿舍中有按照宿舍楼号、房间号、床位号、学号的四种排序方法。在此只列

出按照宿舍楼号进行排序的选择排序算法。

int sort\_select(Stu \*head,int n)//选择排序

{ Stu \*p,\*q;

int lh;

switch(n)//功能选择

{ system("cls");

for(p=head;p!=NULL;p=p->next)//嵌套循环

{ for(q=p->next;q!=NULL;q=q->next)

{ if(p->lh>q->lh)

{ lh=q->lh;q->lh=p->lh;p->lh=lh;}}}

output(head);

system("pause");

return 1;

}

3.3信息查询模块

（1）功能

该模块采用顺序查找法对学生信息进行查询，寝室管理人员可以根据不同情况的需要，对已进行排序的数据文件按关键字（姓名、学号、寝室号）进行查找，采用顺序查找数据不需要有序，也不限制存储方式，所以在本宿舍管理系统中查找使用顺序查找。

（2）算法描述

顺序查找算法是按照序列原有顺序对数组进行遍历比较查询的基本查找算法，让关键字与队列中的数从最后一个开始逐个比较，直到找出与给定关键字相同的数为止。对于任意一个序列以及一个给定的元素，将给定元素与序列中元素依次比较，直到找出与给定关键字相同的元素，或者将序列中的元素与其都比较完为止。顺序查找实现简单，所以本系统采用了顺序查找算法对学生信息数据文件进行查找操作。

（3）具体语言实现

在本系统中有按照宿舍楼号查找、按照房间号查找、按照床位号查找、按照学号查找的四种查询方法。由于代码过长，在此只列出按照宿舍楼号进行查询的顺序查找算法。

int find\_select(Stu \*head,int n)//按学号从小到大查找（采用顺序查找）

{ Stu \*p;

switch(n){

system("cls");//清空

p=head;

int a;

printf("\n\t\t\t要查找的宿舍楼号为：");//输入要查找的宿舍号

......

while(p!=NULL){

if(p->lh==a)

printf("%3d%20d%20d%30s%20s%20d%20s%20s\n",p->lh,p->roomnum,p->bednum,p->xy,p->bj,p->num,p->name,p->sex);

p=p->next;}

system("pause");//暂停

return 1;

}

3.4信息删除模块

（1）功能

由于会有学生因为不同的原因会离开学校，管理员需要及时更新宿舍管理系统的数据文件，以免造成空床浪费，而新生无法入住的情况发生。管理员可以通过查找该学生的学号调取改学生的信息，确认删除该学生信息后，进行对其学生信息数据文件的删除。由于本系统采取的是线性链表结构，已知该链表的表头，写出一个删除某一个节点的算法，要求先找到此节点，然后删除。

（2）算法描述

本系统中学生信息存储的方式是以线性表的链式存储方式进行存储的，删除某个学生信息，从删除元素位置开始遍历到最后一个元素位置，分别将它们都向前移动一个位置，表长减1。

（3）具体语言实现

Student \*del(Stu \*head)

{ int x,y;

printf("\n\t\t\t1.删除某个学生信息");

printf("\n\t\t\t2.删除全部学生信息\n");

printf("\n\t\t\t选择1或2：");

scanf("%d",&x);

if(x==1)

{ printf("\n\t\t\t请输入将要删除学生的位置：");

scanf("%d",&y);

Stu \*p,\*q;

int i;

if(y-1==0)

{head=head->next;}

else{

p=head;

for(i=1;i<y&&p->next!=NULL;i++)

{p=p->next;}

q=p->next;

p->next=q->next;}

printf("\n\t\t\t删除成功！\n");

return head;}

else if(x==2){

head=NULL;

printf("\n\t\t\t删除成功！\n");

return head;}

else{

printf("输入错误，返回主菜单！\n");

return head;}}

if(i>=l.length)

{printf("该学生不存在\n");}

if(k>=0)

{l.length--;}

fflush(stdin);

printf("\n");

printf("是否继续删除?<y/n>:");

scanf("%c",&ch);

system("cls");

if(ch=='y'){Delete(l);}

else{ system("cls");}

}

4 时间复杂度分析

本系统的主要功能是对宿舍信息进行管理，采用的是比较简单的线性表的链式存储结构。算法的时间复杂度是一个函数，它定性描述了该算法的运行时间。这是一个关于代表算法输入值的[字符串](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6%E4%B8%B2/1017763" \t "https://baike.baidu.com/item/_blank)的长度的函数。线性表的链式存储结构的算法思想简单明了，容易转换成相对应的程序源码，它们的时间复杂度为T(n)=O(n)。在本系统中，对数据的排序采用了选择排序算法，其时间复杂度为T(n)= O(n ^2)。但是在查找、删除和插入学生信息时采用顺序查找的方法，找到相对应的记录，则可知其时间复杂度为T(n)=O(n)。

5 系统测试

5.1 测试数据

开始录入的数据：

宿舍楼号 房间号 床位号 学院 班级 学号 姓名 性别

13 359 4 计控 软件161 2016021045 江雪 女

13 359 5 计控 软件162 2016021046 赵哥 女

22 526 4 轻纺 染整161 2016034126 大饼 男

13 201 2 通信 电子166 2016124189 圆子 女

数据排序：

按宿舍楼号 按房间号 按床位号 按学号

插入的数据

宿舍楼号 房间号 床位号 学院 班级 学号 姓名 性别

16 308 6 计控 计本162 2016021137 铁牛 男

查找数据：

13（按宿舍楼号） 359（按房间号） 4（按床位号） 计控（按学院）

软件161（按班级） 2016021045（按学号） 江雪（按姓名） 女（按性别）

修改数据：

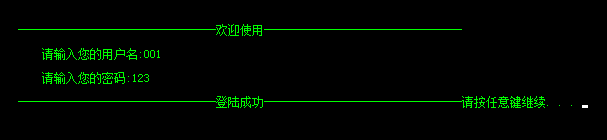
2016021045（按学号）

删除数据：

2（按位置）

5.2 测试输出结果

进入系统后，系统将显示登录界面，正确输入登录用户名和密码后，系统提示登陆成功，按任意键即可进入主菜单界面，进而选择对学生信息的一系列操作，界面较为简单，方便管理员的操作，浅显易懂如图5-1所示。

图5-1 欢迎界面图

若用户名或密码连续三次输入错误，将自动退出系统，如图5-2所示。根据对代

码的仔细研究，发现在操作菜单函数中有代码如下所示（由于代码过长只截取主要部分）,采用for循环。

for(i=1;i<4;i++)

{ ......

if ((strcmp(userName,"001")==0) && (strcmp(password,"123")==0))

/\*验证用户名和密码\*/

{printf("\n 登陆成功");return; }

else{

if(i<3){

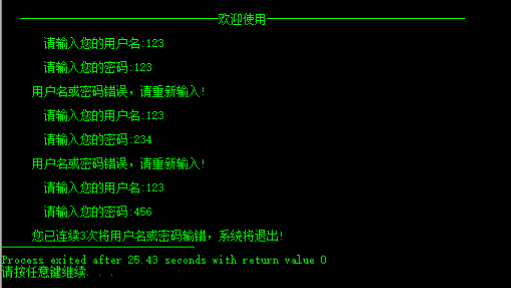
printf("\n 用户名或密码错误，请重新输入!\n"); }

else{

printf("\n 您已连续3次将用户名或密码输错，系统退出!");

exit(0);}}

}

图5-2 用户名或密码错误界面图

用户名和密码正确，按任意键后便进入线性表的主菜单界面，输入数据在0—9之间时，分别进入各小菜单。在调试过程中，当输入n后，出现运行错误，输入数据不存在，返回主菜单界面，如图5-3所示。此处选用do-while循环，代码如下所示（简写）。

do { system("cls");

printf("... ");

scanf("%d",&a);

}while(a<0||a>9);

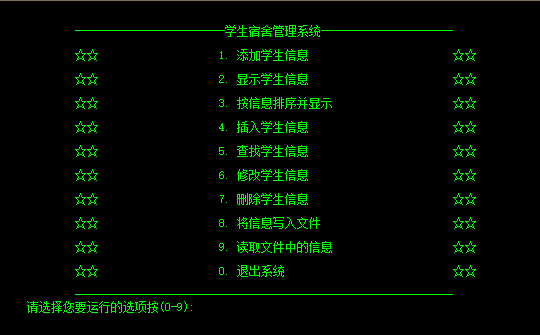
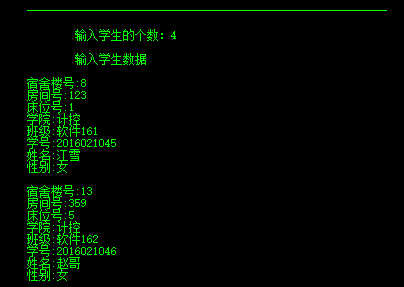


图5-3 主菜单界面图

在界面中输入数据1，开始创建学生信息表，选择输入学生信息的个数，依次输入学生信息。每当输入一个信息后，系统会继续弹出下一名学生的信息输入，继续输入直到所输入学生信息个数全部输入完成，按任意键返回主菜单界面，如图5-4所示。

 图5-4 学生信息录入图

管理员在主菜单界面选择输入数据2，则显示刚刚依次输入的学生信息。显示输入信息是为了方便管理员检查信息是否输入正确，若检查出输入错误，则可以及时改正，如图5-5 所示。



图5-5 学生信息显示界面图

输入数据3选择排序方式，进入选择排序方式界面，通过输入不同的数字对系统进行不同的操作如图5-6所示。

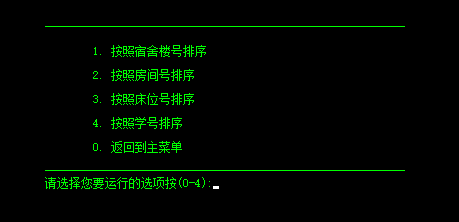


图5-6 选择排序方式界面图

管理员选择对学生数据信息进行按宿舍楼号排序，按照宿舍楼号按从小到大排列，结果如图5-7所示。

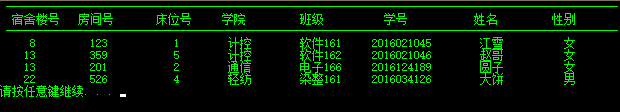


图5-7 按宿舍楼号排序图

管理员选择对学生数据信息进行按房间号排序，按照房间号按从小到大排列，结果如图5-8所示。

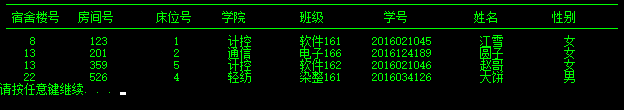


图5-8 按房间号排序图

管理员选择对学生数据信息进行按床位号排序，房间号按从小到大排列，结果如图5-9所示。

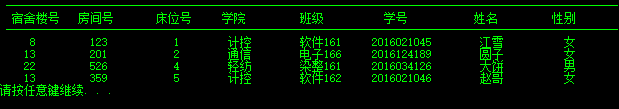


图5-9 按寝室号排序图

管理员选择对学生数据信息进行按学号排序，寝室号按从小到大排列，结果如图5-10所示。

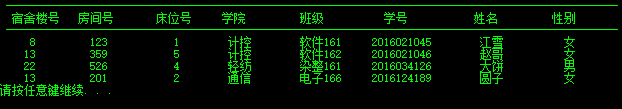


图5-10 按学号排序图

排序完成后，选择0返回到主菜单界面，至此所有学生的宿舍信息排序完成，宿舍管理系统按照管理员所选择的排序方式显示学生的宿舍信息，返回宿舍管理系统主菜单进行其他操作。

由于新同学入住，管理员选择插入学生信息，进行插入学生宿舍信息数据操作，输入宿舍楼号后下一行弹出房间号，输入房间号后弹出床位号，依次输入学院、班级、学号、姓名、性别等信息，如图5-11所示。

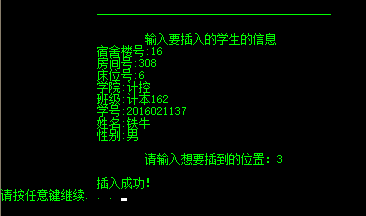


图5-11 插入学生信息图

由于管理需要，管理员需要查找学生，选择查找信息进入选择查找方式界面，由于学生信息量庞大，信息内容复杂，又考虑到部分信息遗忘或丢失的情况，本系统为了方便宿舍管理人员的查找，并且又根据学生所有的信息特点做出了八种信息查找方式，这些查找方式分别是按照宿舍楼号查找、按照房间号查找、按照床位号查找、按照学员查找、按照班级查找、按照学号查找、按照姓名查找、按照性别查找，如图5-12所示。

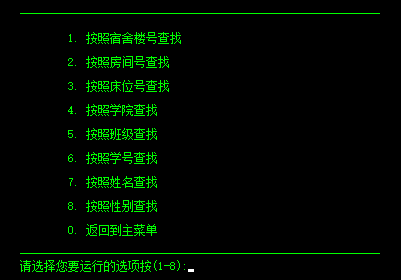


图5-12 选择查找方式界面图

管理员选择查找方式，由于查找方式较多，代码过长，此处就不一一例举，此处取其中一个查找方式演示，例如按照学院查找，选择查找信息后，弹出“要查找的学院为：”，输入学院名称后，显示所有该学院的学生宿舍信息，排序方式按学生信息录入的先后顺序为准，如图5-13所示。

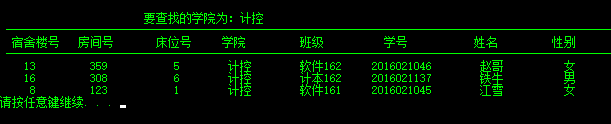


图5-13 按照学院查找图

查找结束后，管理员选择返回主菜单。

由于所输入的信息可能有误，所以设置修改学生信息选项，管理员选择修改学生信息，输入要修改学生之前录入的学号，方便依次修改学生信息方式，如图5-12所示。

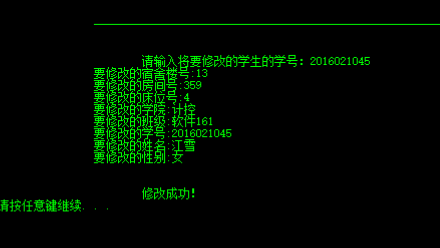


图5-14 修改学生信息图

由于学生不同原因离校，管理员要及时删除离校学生信息，进入删除界面后，系统提示删除某个学生信息还是全部学生信息，若管理员选择全部，则全部学生信息删除，若管理员选择删除某个信息则需要输入学生信息所在位置，即可显示出要删除的该学生信息，便于管理员确认该学生信息并删除，弹出是否确认删除该学生信息，选择“是”，则该学生宿舍信息删除成功，删除操作到此结束，按任意键返回主菜单界面，如图5-15所示。

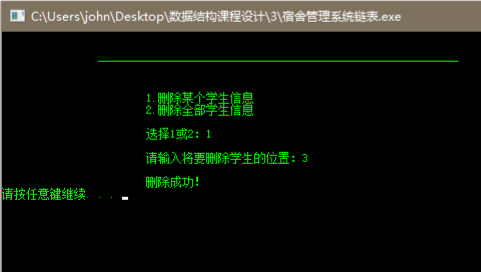


图5-15 删除学生信息图

为了方便学生信息的保存和管理员之间的信息文件传送，需要将学生信息写入文件，将学生信息写入文件的好处是方便学生宿舍信息的保存，有需要时可以进行学生信息的公示以及管理员和老师或者学校管理层之间学生信息的传送和对比，如图5-16所示。

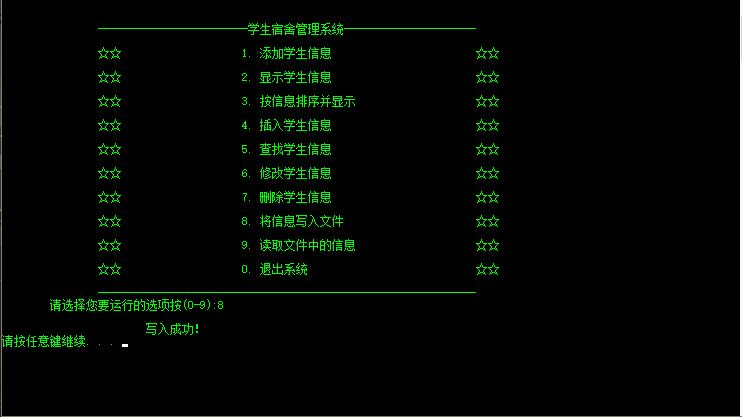


图5-16 信息写入文件图

管理员再次进入系统查询学生信息时，直接读取文件中的信息就可以查看之前录入的所有学生的宿舍信息，再次进行添加、插入、修改学生宿舍信息时，可以同时存入之前的文件“学生宿舍信息”中，方便学生信息的修改、插入和获取，如图5-17所示。

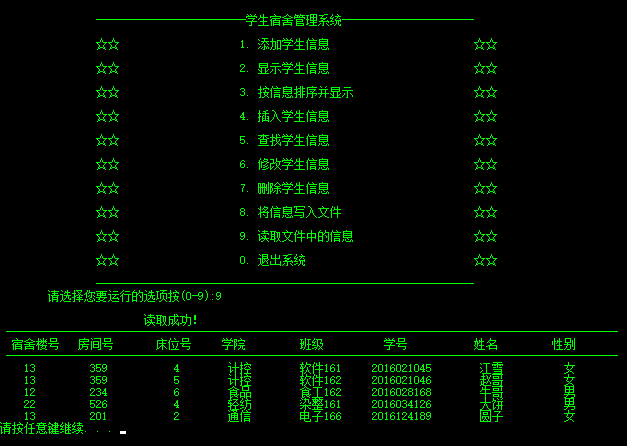


图5-17 读取文件信息图

当所有操作完成后，管理员选择退出系统。

6 总结

本系统是为了让所有宿舍管理人员能从繁重的工作中解脱出来，为了方便宿舍管理人员实现宿舍管理查询而开发的，具有信息录入、信息显示、信息排序、信息查询、信息修改、信息插入和信息删除的功能。运用了之前学过的线性表的链式存储结构来存储数据，选择排序算法对数据文件进行排序，并且在排序后的基础上，运用了顺序查找算法对所需的数据文件进行查找。

通过本次数据结构课程设计，我学到了很多，我觉得很满足，也很有成就感，许多曾经自己不知道的知识，现在知道了并能熟练的掌握了一些，比如：

（1）fflush(fp)函数，刷新缓冲区，将缓冲区的内容写入文件

（2）system("cls")函数，将屏幕先前显示的内容清理掉。

（3）当scanf()输入字符串时不能带空格，否则不能输出后面的字符，而gets()能输入带空格的字符串。

（4）以前不知道怎么把各个函数编排在一起，不能形成一个总体模块，现在经过实践与调试，已经能组装好各个函数模块，使它们实现各自的功能。

众所周知，要学好编程方面这一块，数据结构是必可少的，它里面有许多经典的算法和思想，是解决问题的必需品。很惭愧，我掌握的知识很少，特别是树和图，只知道皮毛，目前还不能用这两方面的知识编程。在本系统中，我运用了数据结构中的链表知识，运用单链表表完成了宿舍管理系统软件，能够实现动态的存储。在这次课程设计中，经过自己的编成和调试，我成功的完成了这次作业，虽然它的功能还不是很多，也不是很完善，但我还是非常的兴奋，因为，它给我增添了编程方面的信心，使我更有信心去学习编程。