

第 11 章 学生信息管理系统

（视频讲解：1 小时）

本章我们将开始尝试做一个学生管理系统，第一部分是系统需求讲解，接着第二部分是设计，设计仅仅是一个参考，内存中可以用链表存储，也可以用哈希，或者用红黑树都可以，并不限制。第三部分给出了一些函数，便于大家在代码编写时进行模块划分，减少重复代码。

11.1 系统功能描述

该系统管理学生的课程信息。系统提供账户的管理和学生信息的管理功能。系统提供的功能主要有 学生信息的查询、修改、增加、删除，账户信息的查询、修改、增加、删除。（如图 11.1-1）

```
*****Student Information Management System*****
1.search      student information
2.add student information
3.update student information
4.delete student      information
5.add user      account
6.update user      account
7.delete user      account
8.search user      account
9.exit
select a number: _
```

图 11.1-1

登录管理系统的账户分为管理员账户和普通账户。

普通用户只具有查询学生信息的功能, 如图 11.1-2。

```
*****Student Information Management System*****
1.search      student information
9.exit
select a number: _
```

图 11.1-2

管理员用户具有系统提供的所有功能。

主要用到的数据结构有：链表（内存中存储账号和学生信息）。

账号和学生信息是一个结构体。

```
typedef struct usr_account_tag
{
char usr_name[USR_NAME_LEN];
```

```

char usr_pwd[USR_PWD_LEN];
USR_ROLE  usr_role ;
struct usr_account_tag *pNext_Usr_Account ;
}USR_ACCOUNT, *pUSR_ACCOUNT;

```

```

typedef struct usr_infor_tag
{
    int usr_id ;
    char usr_name[USR_NAME_LEN] ;
    int usr_course_id ;
    double usr_course_score;
    struct usr_infor_tag *pNext_Usr_Infor ;

}USR_INFOR,*pUSR_INFOR;

```

系统数据主要存储在三个文件：configure.txt usr_account.txt
usr_infor.txt .

Usr_account.txt 主要用于存放用户账户信息

Usr_infor.txt 主要用于存放学生信息

Configure.txt 主要用于存放配置信息：usr_account.txt,usr_infor.txt
的位置信息。

项目需求不理解的同学可以加入前言 QQ 群，通过视频辅助进行理解

11.2 系统模块设计

a. 预处理模块

系统在启动时会根据配置文件里的内容去相应文件里去加载账户信息和
学生信息。

b. 登陆模块

```
enter usr_name:
```

```
enter usr_password:
```

```
enter usr_password:*****
```

输入用户名和密码，输密码的时候用“*”代表用户当前输入的内容。
输入回车代表结束。

当用户输入完用户名和密码后，系统会拿它去跟内存中的用户名和密码

匹配。如果匹配成功则进入系统菜单界面。如果匹配不成功则会直接退出程序。实现方法是采用

```
#include<conio.h>    char c=getch()
```

getch() 输入时, 不会显示在屏幕上, 读到一个字符后, 就输出一个*, 当读到的是\b 字符, 如何实现退格呢? 先输出一个\b, 再输出一个空格, 再输出一个\b, 即使用 printf("\b \b");

c. 系统功能菜单显示

根据用户名的权限不同显示不同的功能, 普通用户只具有查询学生信息的功能。

当用户选择查询学生信息功能时会根据用户权限的不同显示不同功能的查询功能。当前用户为普通用户时, 显示如下:

```
1. search by name
2. search by id
3. return
_
```

分为按姓名查询和按学号查询。3 菜单返回上一层。

当前用户为管理员用户时, 显示为

```
0. search all
1. search by name
2. search by id
3. return
```

多了一个功能。

系统提供的其他功能只有在管理员权限下才能使用, 见图 11.2-1。

```
1.search      student information
2.add  student information
3.update student information
4.delete student      information
5.add  user   account
6.update      user   account
7.delete      user   account
8.search      user   account
9.exit
```

图 11.2-1

菜单 2 为添加学生信息, 管理员从标准输入输入学生的详细信息, 系统会把学生信息按照学号的大小插入到内存的链表中, 并同时把链表保存到相应文件中。去。(注意: 学生的学号是唯一的, 如果两学生的学号相同会插入失败)

菜单 3 为修改学生的信息。管理员可以根据学号修改学生的其他信息 (学号不能修改)。并同时修改完的信息保存到文件。

菜单 4 为删除学生的信息, 管理员可以根据学生的学号删除相应学生的信息。并同时把删除后的学生信息保存到文件中。

菜单 5 是增加账号。所增加的账号名不能与已存在的冲突。增加账号完毕后同时把增加后的信息保存到相应的文件中。

菜单 6 更新账号信息。同时把信息保存到相应文件中。

菜单 7 删除用户账户。管理员根据输入的账户名删除相对应的账户。并把删除后的信息保存到相应的文件中。

菜单 8 查询用户账号信息。

菜单 9 退出系统。并同时把信息保存到文件中。

11.3 系统用到的相关函数，仅供参考

```
void system_init(const char* config_file_name, pUSR_ACCOUNT *pUsr_Account,
pUSR_INFOR *pUsr_Infor);
```

系统初始化函数。初始化内存中的学生信息链表和账号列表。

```
USR_ROLE role_confirm(pUSR_ACCOUNT pUsr_Account, USR_ACCOUNT_NUM
usr_account_num);
```

根据输入的用户名和密码检查是否合法并确认当前账号的权限(管理员或普通用户)

```
SYSTEM_MENU_TYPE show_system_menu(USR_ROLE role);
```

显示系统菜单，返回当前用户所选择的菜单

```
SYSTEM_MENU_TYPE show_search_menu(USR_ROLE role);
```

当前用户选择查询功能时根据权限显示具有不同功能的查询功能的菜单

```
void search_usr_infor_by_id(pUSR_INFOR pUsr_Infor_List, SEARCH_ID search_id);
```

根据当前用户所输的学号查询相应的学生信息。

```
void search_usr_infor_by_name(pUSR_INFOR pUsr_Infor_List, char*
search_name);
```

根据当前用户所输的姓名查询相应的学生信息。

```
pUSR_INFOR add_usr_infor(pUSR_INFOR *ppUsr_Infor_List, USR_INFOR_NUM *
usr_infor_num);
```

添加学生信息，返回指向插入的学生的指针

```
pUSR_INFOR update_usr_infor(pUSR_INFOR pUsr_Infor_List, SEARCH_ID
search_id);
```

修改学生的信息，返回指向修改后的学生的信息的指针

```
pUSR_INFOR delete_usr_infor(pUSR_INFOR *ppUsr_Infor_List, USR_INFOR_NUM
* usr_infor_num, SEARCH_ID search_id);
```

删除学生的信息，是返回所删除的学生的下一个学生的指针

```
void search_usr_account(pUSR_ACCOUNT pUsr_Account_List, char*
search_name);
```

根据当前账号输入的账号名查询相应的账号信息。

```
pUSR_ACCOUNT add_usr_account(pUSR_ACCOUNT* ppUsr_Account_List,
```

USR_ACCOUNT_NUM* usr_account_num);

增加账号信息，返回指向当前账号的指针

pUSR_ACCOUNT update_usr_account(pUSR_ACCOUNT pUsr_Account_List,char* search_name);

修改账号信息返回指向当前修改账号的信息

pUSR_ACCOUNT delete_usr_account(pUSR_ACCOUNT *ppUsr_Account_List,USR_ACCOUNT_NUM* usr_account_num,char* search_name);

删除账号信息，返回指向下一个账号的指针

void save_data_to_file(char * config_file_name ,pUSR_ACCOUNT pUsr_Account_List,pUSR_INFOR pUsr_Infor_List);

将学生信息和账号信息保存到文件里去。

提供以上函数的目的是为了大家在编写项目时，注意每个功能的封装，不要写重复代码。