2.28

1、完成数据查询（至少两个单项查询、一个组合查询）功能  
2、完成数据修改功能  
3、若昨天的数据输入、输出、存储等环节未实现，请今日补上  
4、晚上22:00前提交工作日志，包括完成功能、重点代码实现、心得体会等。

1.查询

登录： BLL层用域名查该账户结构体

struct user\_list \* query(char \* username)

{

struct user\_list \* t;

t=userlist->next;

while(t!=NULL)

{

if(strcmp(username,t->user\_name)==0)

{

return t;

}

t=t->next;

}

return NULL;

}

今日内容是完善登陆系统和管理员系统，管理员端查询与上类似，明天再补

2.实例：注册

void logon\_UI(int step) //注册；此处注册权限为4

{

printf("\n\t\t\t\t\t\t\t\t\t\t 请输入账号（需以字母开头且包含7-19个字符）：");

char username[1000],password[1000];

scanf("%s",username);

printf("\n\t\t\t\t\t\t\t\t\t\t 请输入密码（长度为7-19个字符）：");

scanf("%s",password);

int s1=logonuser\_BLL(username,password);

if(s1==1)

{

system("cls");

printf("\n\t\t\t\t\t\t\t\t\t\t 账号或密码长度不符合要求，请重新输入\n");

logon\_UI(0);

return;

}

if(s1==2)

{

system("cls");

printf("\n\t\t\t\t\t\t\t\t\t\t 账号已存在，请重新注册\n");

logon\_UI(0);

return ;

}

if(s1==0)

{

user\_add(username,password,4);

}

system("cls");

printf("\n\t\t\t\t\t\t\t\t\t\t 注册成功,前往登录\n");

login\_UI(0);

/\* struct user\_list \* query(char \* username)\*/

return;

}

Ui层输入账号密码，bll层判断合法，合法加入链表并重新写入文件

Bll层

void user\_add(char \*username,char \*password,int jur)

{

struct user\_list \* t=(struct user\_list \*)malloc(sizeof(struct user\_list));

strcpy(t->user\_name,username);

strcpy(t->user\_password,password);

t->jurisdiction=jur;

t->next=userlist->next;

userlist->next=t;

user\_printf();

return ;

}

账号密码不需要排序，直接头插法省遍历时间

Dal层

void user\_printf()

{

FILE \*fp;

fp=fopen("./DAL/user\_list.ini","w+");

struct user\_list \* t;

t=userlist->next;

while(t!=NULL)

{

fprintf(fp,"%s %s %d\n",t->user\_name,t->user\_password,t->jurisdiction);

t=t->next;

}

fclose(fp);

return ;

}

心得：今日重写了登陆系统代码，主要在于昨天未分清逻辑层与数据层关系，规划了29日用链表实现按播出时间，票房排序的思路：在插入时直接整成有序链表。

错误，在程序编译时出现了递归包含错误，百度解决：加预定义并判断是否已经定义。