

# C 语言程序设计 案例实践

李丹程 刘莹 那俊 著

清华大学出版社

# 目 录

预备知识——Visual Studio 6.0 开发环境 .....	1
一、Visual C++ 6.0 的安装和启动 .....	1
二、编辑一个C源程序 .....	2
三、编译、链接和运行C程序 .....	6
四、C程序的调试 .....	8
<b>第一章 学生信息管理 .....</b>	<b>11</b>
一、实践目的 .....	11
二、基本要求 .....	11
三、算法分析 .....	11
四、参考代码 .....	14
五、代码测试 .....	37
六、思考题 .....	38
<b>第二章 旅馆信息管理 .....</b>	<b>39</b>
一、实践目的 .....	39
二、基本要求 .....	39
三、算法分析 .....	39
四、参考代码 .....	42
五、代码测试 .....	54
六、思考题 .....	55
<b>第三章 餐厅信息管理 .....</b>	<b>56</b>
一、实践目的 .....	56
二、基本要求 .....	56
三、算法分析 .....	56
四、参考代码 .....	58
五、代码测试 .....	66
六、思考题 .....	67
<b>第四章 职工信息管理 .....</b>	<b>68</b>
一、实践目的 .....	68
二、基本要求 .....	68
三、算法分析 .....	69
四、参考代码 .....	70
五、代码测试 .....	82
六、思考题 .....	85
<b>第五章 设备信息管理 .....</b>	<b>86</b>
一、实践目的 .....	86
二、基本要求 .....	86
三、算法分析 .....	87
四、参考代码 .....	89
五、代码测试 .....	98
六、思考题 .....	100
<b>第六章 图书信息管理 .....</b>	<b>101</b>

一、实践目的 .....	101
二、基本要求 .....	101
三、算法分析 .....	101
四、参考代码 .....	103
五、代码测试 .....	108
六、思考题 .....	109
<b>第七章 医院信息管理.....</b>	<b>110</b>
一、实践目的 .....	110
二、基本要求 .....	110
三、算法分析 .....	110
四、参考代码 .....	112
五、测试代码 .....	119
六、思考题 .....	120
<b>第八章 超市信息管理.....</b>	<b>121</b>
一、实践目的 .....	121
二、基本要求 .....	121
三、算法分析 .....	121
四、参考代码 .....	122
五、代码测试 .....	127
六、思考题 .....	129
<b>第九章 票务信息管理.....</b>	<b>130</b>
一、实践目的 .....	130
二、基本要求 .....	130
三、算法分析 .....	130
四、参考代码 .....	133
五、代码测试 .....	136
六、思考题 .....	137
<b>第十章 超级终端仿真.....</b>	<b>138</b>
一、实践目的 .....	138
二、基本要求 .....	138
三、算法分析 .....	138
四、基本步骤 .....	139
五、参考代码 .....	142
六、代码测试 .....	144
七、思考题 .....	144

## 预备知识——Visual Studio 6.0 开发环境

本书以 Visual C++ 6.0 作为 C 源程序的实践开发环境，本章将首先介绍 Visual C++ 6.0 环境的基本操作，包括 Visual C++ 6.0 的安装和启动、C 源程序的编辑、运行与调试以及多文件工程的建立和运行方法。

### 一、Visual C++ 6.0 的安装和启动

如果机器上未安装 Visual C++ 6.0，则按照安装向导直接安装，具体步骤此处不再详述。安装过程中建议各位同时安装 MSDN，以便日后自学。

成功安装 Visual C++ 6.0 后，可以在桌面上看到如图 0.1 所示的图标或通过“开始”->“程序”->“Microsoft Visual Studio”找到“Visual C++ 6.0”启动程序。



图 0.1 Visual C++ 6.0 图标

双击 Visual C++ 6.0 图标或者点击“开始”->“程序”->“Microsoft Visual Studio”->“Visual C++ 6.0”即可启动 Visual C++ 6.0 集成开发环境，正常启动开发环境后，可见如图 0.2 所示的主窗口操作界面。

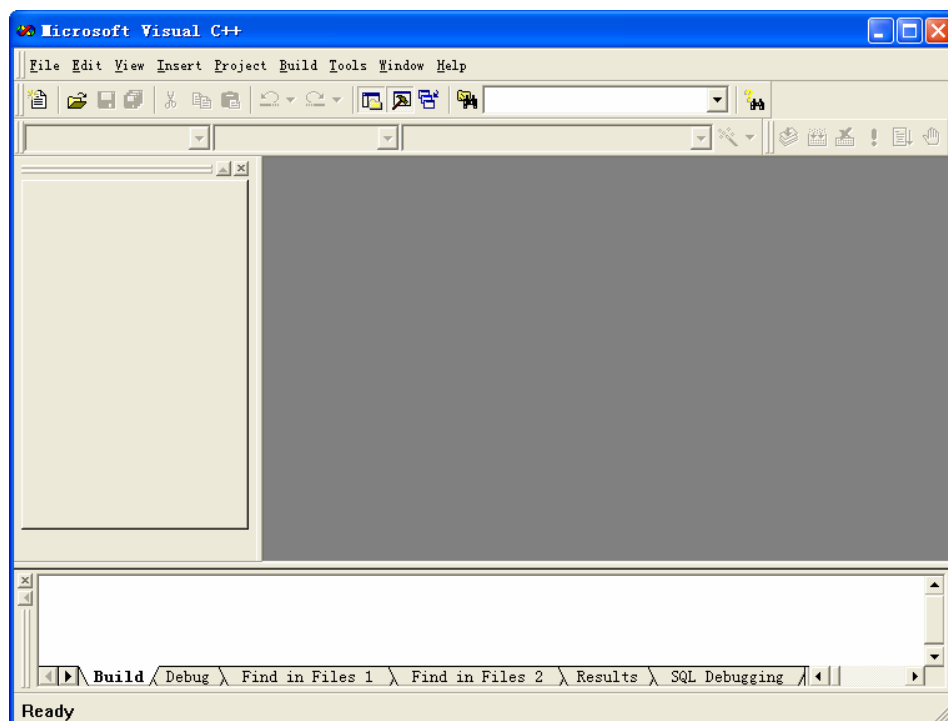


图 0.2 Visual C++ 6.0 主窗口

## 二、编辑一个 C 源程序

编辑一个 C 源程序的前提是新建一个程序或者打开现有的程序。本节首先介绍如何新建一个程序，如何打开现有程序，并在此基础上对文件进行编辑。

### 1、新建一个 C 源程序

#### (1) 新建一个工程

在 Visual C++ 6.0 主窗口的界面中的主菜单栏中单击 File(文件)菜单项，然后单击 New(新建)菜单项，也可按 (Ctrl+N) 进入，如图 0.3 所示。

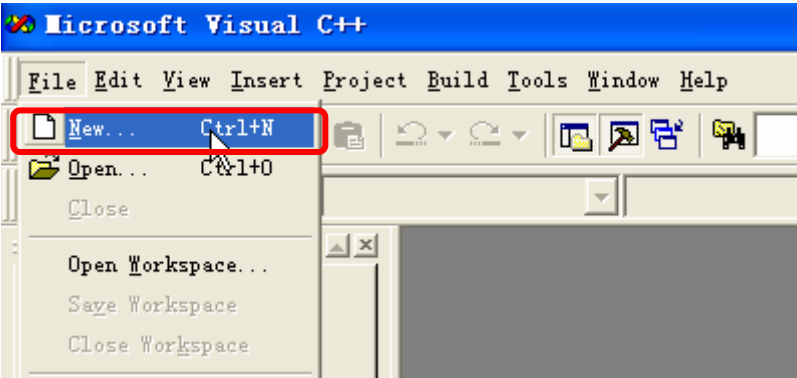


图 0.3 新建工程

屏幕上出现一个 New(新建)对话框，如图 0.4 所示。单击此对话框上方的 Project(工程)标签，在其下方菜单选择“Win32 Console Application”项，输入工程名，设定工程存储路径(Location)然后单击 OK 按钮。

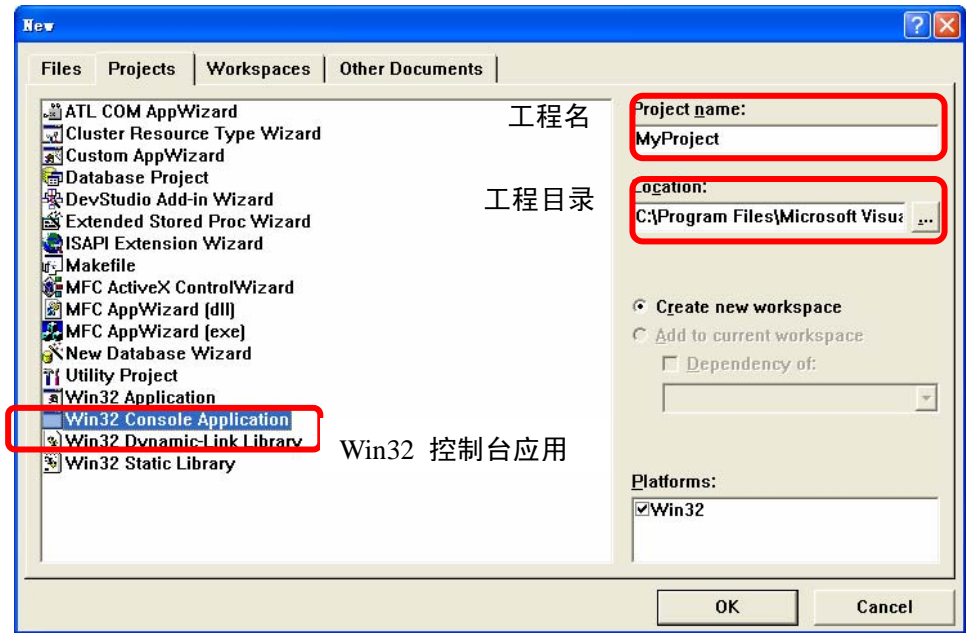


图 0.4 建立工程

单击 OK 后，会弹出如图 0.5 所示的界面，选择 An empty project，单击 Finish。

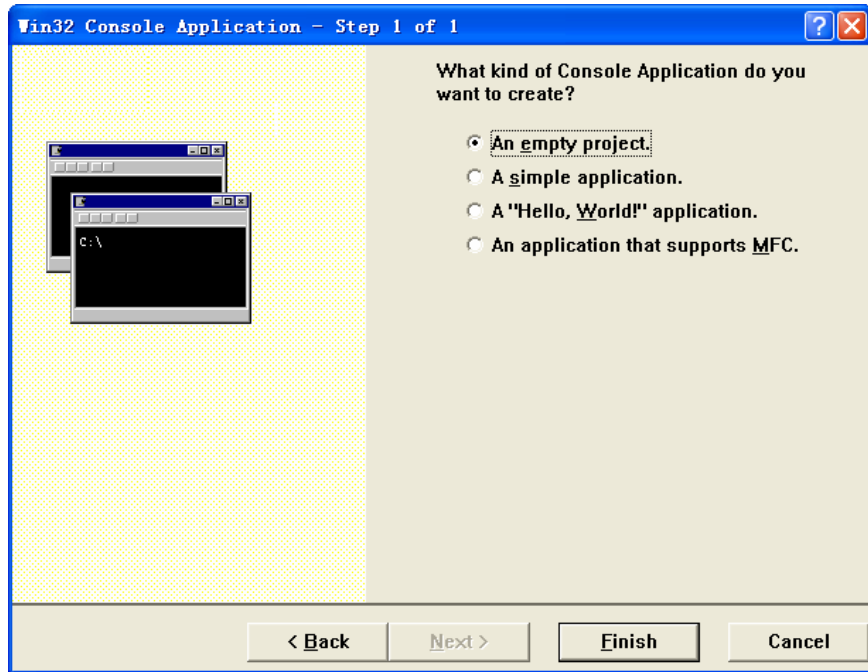


图 0.5 工程类型选择

之后，弹出如图 0.6 所示的界面。注意：窗口上显示了工程文件的路径。

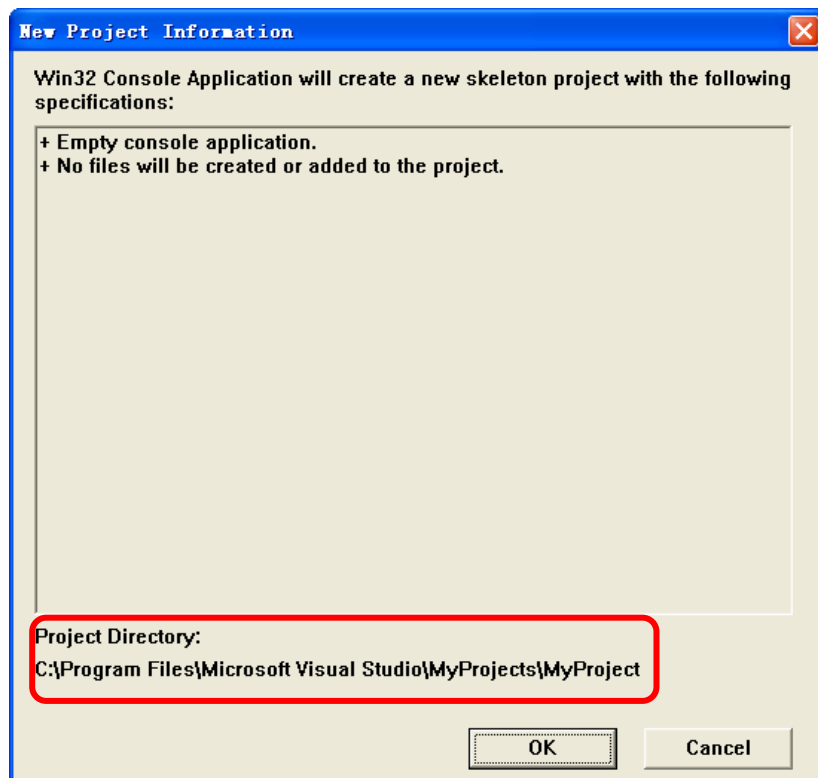


图 0.6 新工程信息

点击 OK，则进入工程编辑窗口，如图 0.7 所示。在工程窗口可以看到 MyProject 工程下只有三个空白的文件夹。

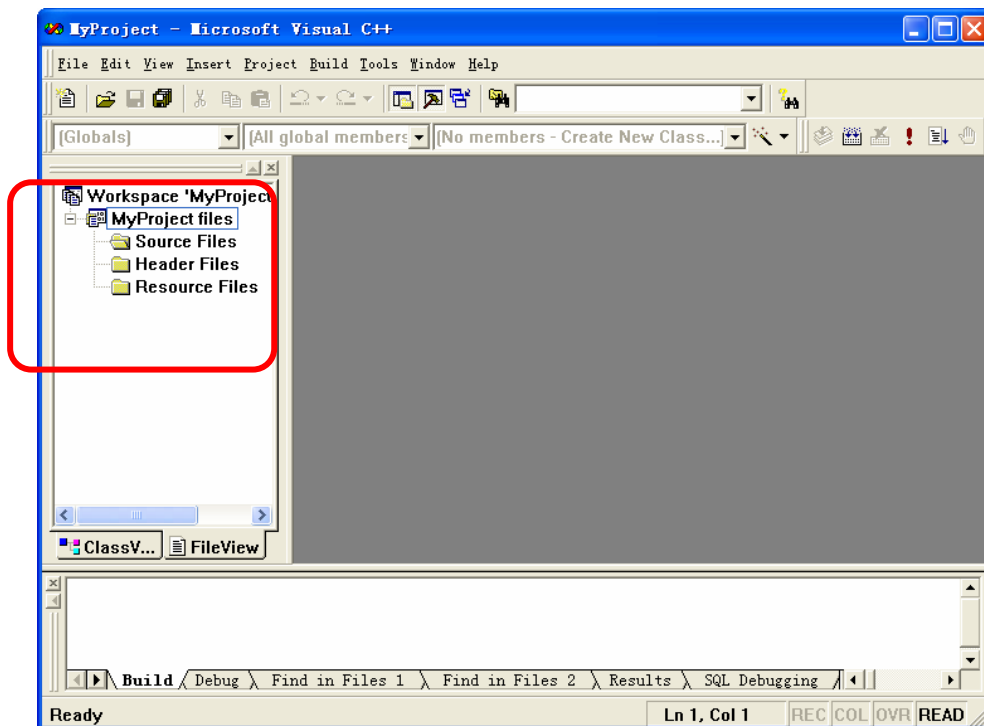


图 0.7 MyProject 工程编辑窗口

此后，就可以在当前工程下创建一个 C 源程序。

## (2) 新建一个 C 源程序

单击 File，选择 New。在新建窗口中，类似新建工程操作（如图 0.8 所示），在 Files 标签选择“C++ Source File”；在 Add to Project 前打钩，选择添加到刚新建的工程 MyProject；在 File 输入框中，输入后缀名是.c 的文件名，单击 OK。

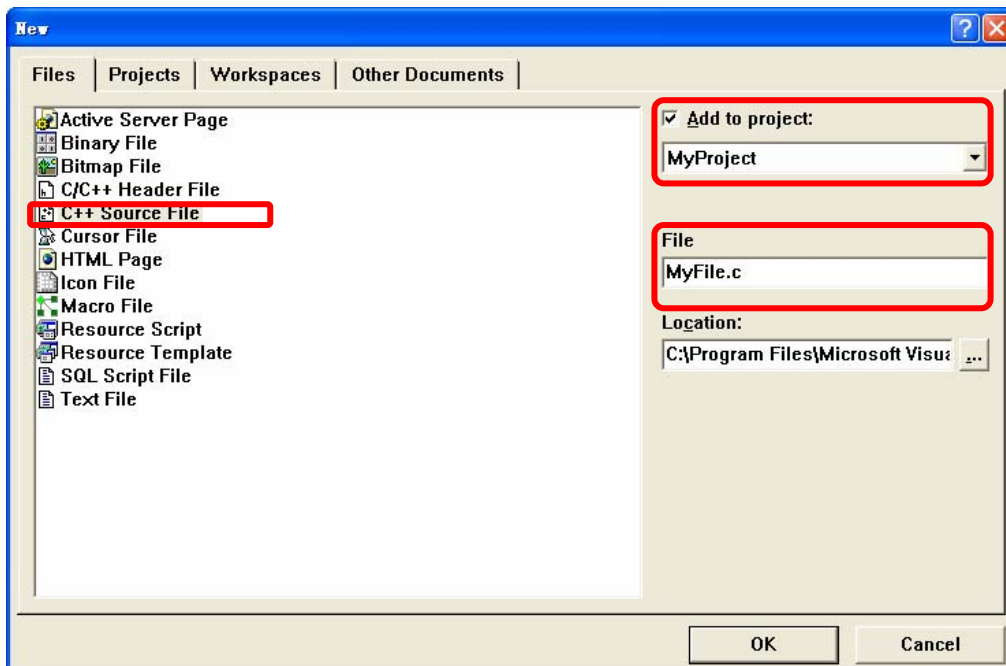


图 0.8 新建 C 源程序

注意：如果在文件名中不显式地输入扩展名.c，则 VC++ 将为文件附上默认扩展名.cpp，并按照 C++ 语言的语法进行检查。由于 C++ 的语法检查要比 C 语言的语法更为严格，因此，建议各位显式地输入文件扩展名.c。

点击 OK 后，可以进入 MyFile.c 的编辑界面，如图 0.9 所示。在窗口的标题栏和工程窗口中都显示出了当前要编辑的文件名字“MyFile.c”。

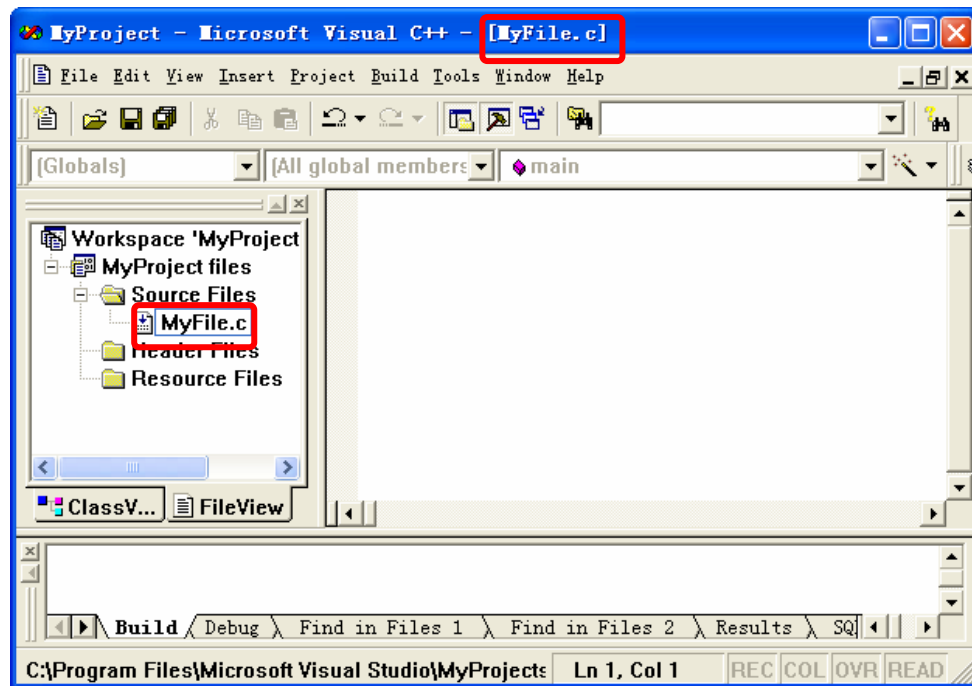


图 0.9 MyFile.c 编辑窗口

此时，程序编辑窗口被激活，可以输入和编辑源程序了。这里，我们编写了一个简单的 C 程序，输出“Hello World!”，如图 0.10 所示。

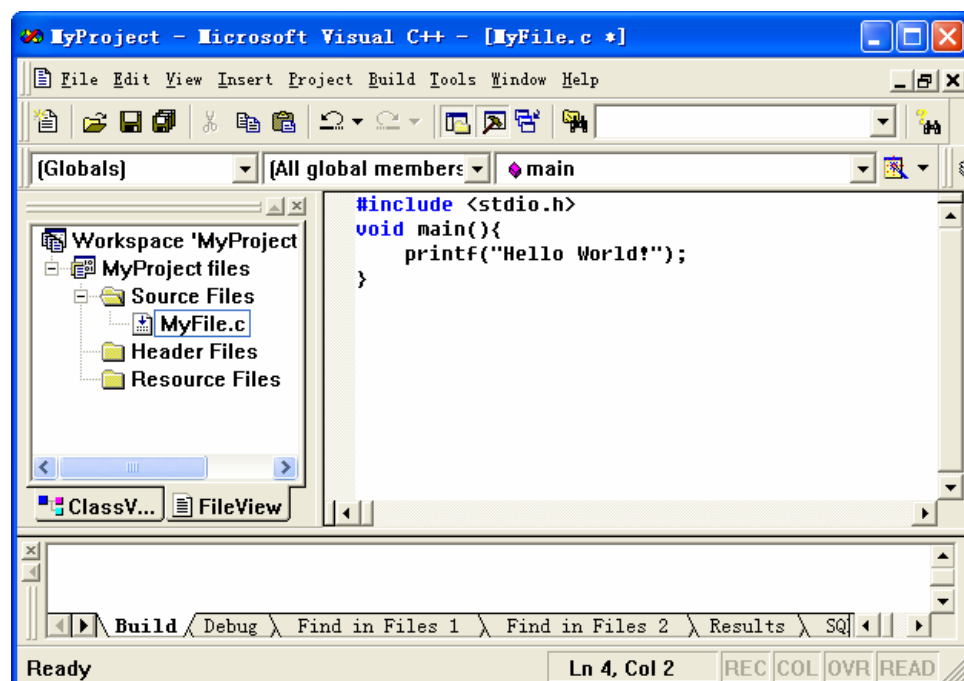



图 0.10 MyFile.c 的编辑



进入编辑状态后,如果对源程序进行了修改且未保存则在标题栏中文件名字后面会出现“\*”提示。选择 File->Save, 点击  按钮, 或者使用快捷键 Ctrl+S 都可以保存文件。保存之后, 标题栏中的“\*”消失。

如果不想将源程序保存到指定的文件中, 可以选择 File->Save As 项, 重新指定文件保存的位置以及文件名字。

## 2、打开一个现存程序

与新建的操作类似, 可以通过 File->Open Workspace 项进入打开工程窗口, 如图 0-11 所示。双击工程文件 (扩展名为.dsw), 即可打开现有的工程, 进入图 0.11 所示的界面。此外, 也可以通过双击工程文件直接打开现有程序。

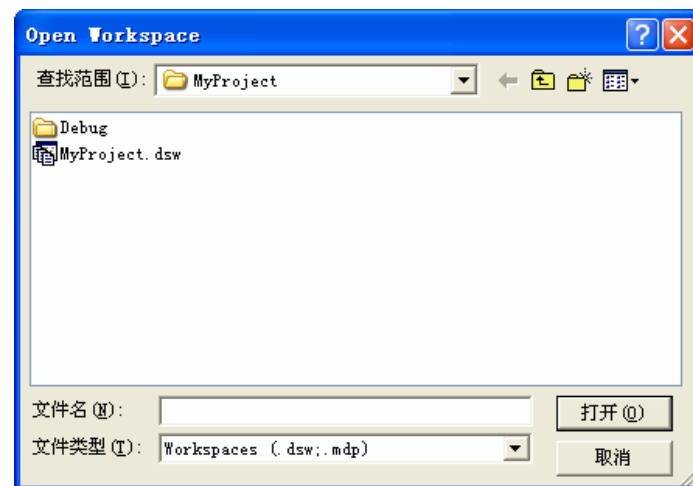


图 0.11 打开工程窗口

## 三、编译、链接和运行 C 程序

### 1、C 程序的编译

编译可以检查程序中是否存在语法错误并生成目标文件 (.obj), 编译结果显示在输出窗口中, 如图 0.12 所示。

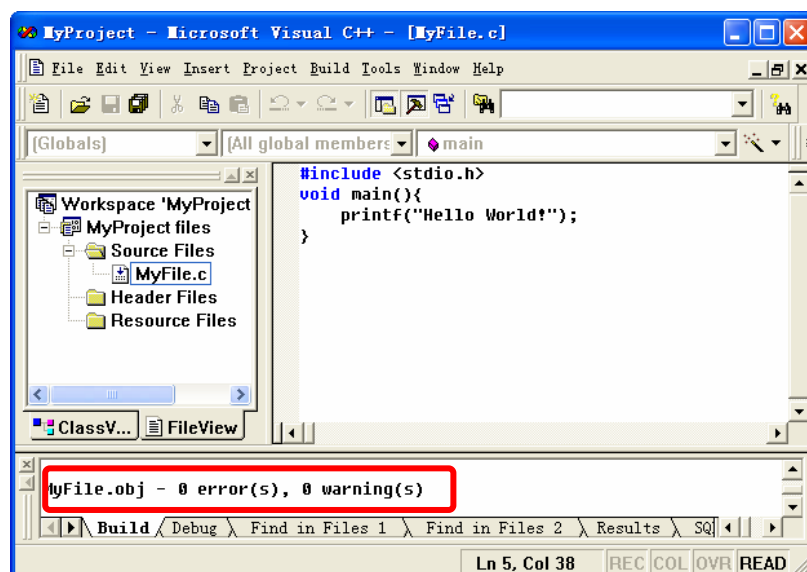


图 0.12 程序编译结果

选择 Build->Compile MyFile.c 即可对 MyFile.c 进行编译。此外，也可以通过点击  按钮或直接按 Ctrl+F7 进行编译。

如果程序中存在语法错误，则可以通过双击错误提示在程序文件中定位到错误所在的代码行。例如，我们把 printf 语句的分号去掉，则编译出现如图 0.13 所示的结果。根据提示框中的提示信息“syntax error: missing ';' before '}'”以及编辑区中的蓝色提示箭头则可以直接定位到错误处进行修改。

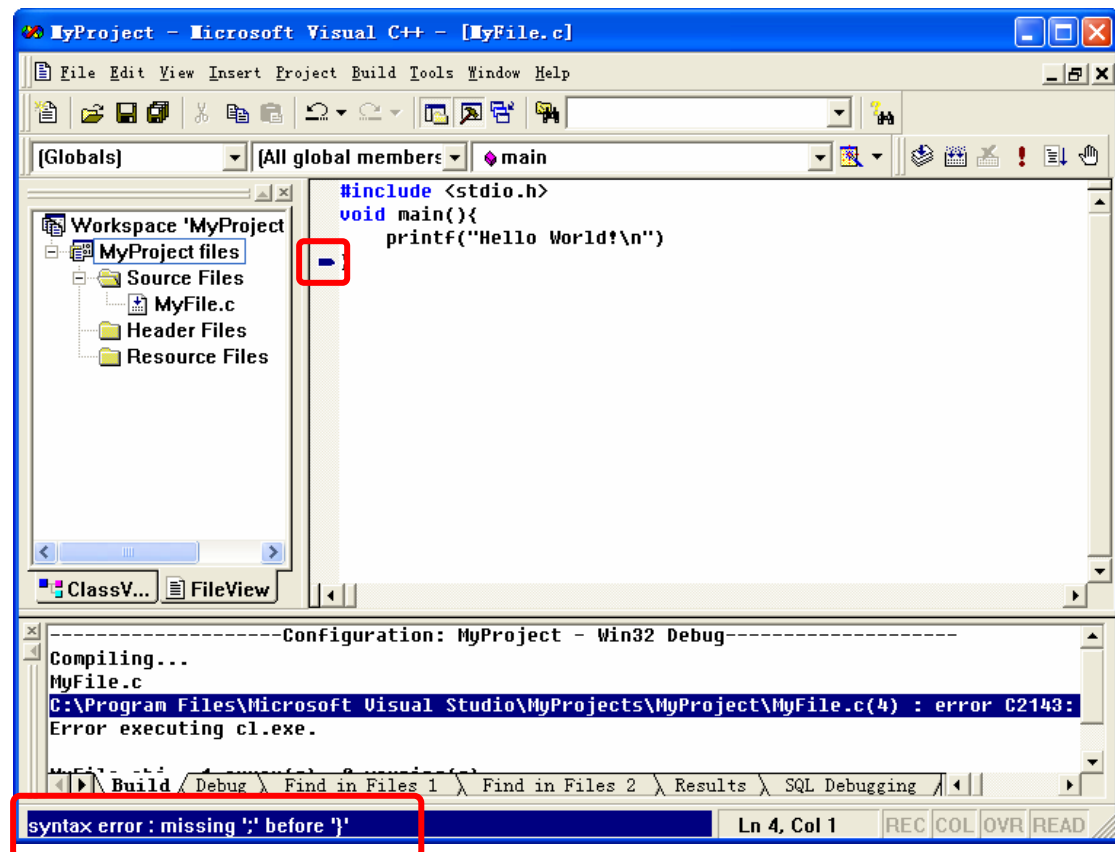



图 0.13 语法错误

值得注意的是，语法错误分为 error 和 warning 两类。error 是一类致命错误，程序中如果有此类错误则无法生成目标程序，更不能执行。warning 则是相对轻微的一类错误，不会影响目标文件及可执行文件的生成，但有可能影响程序的运行结果。因此，建议最好把所有错误（不论是 error 还是 warning）都一一修正。

## 2、C 程序的连接

连接将生成可执行文件（.exe）。选择 Build->Build 选项可以完成程序的连接。此外，也可以通过点击  按钮或通过快捷键 F7 进行连接。类似地，连接后的结果也会出现在输出窗口中，如图 0.14 所示。如果连接失败，则同样会显示出失败的具体原因。

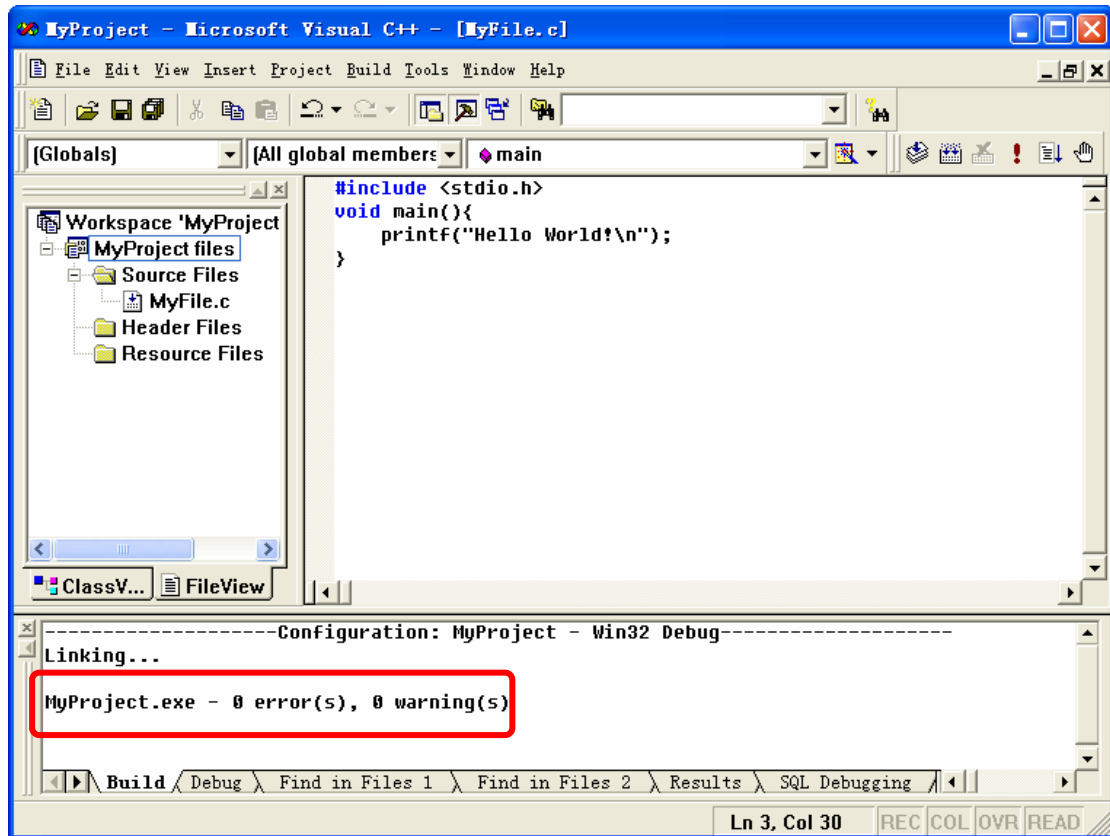


图 0.14 程序连接结果

### 3、C 程序的运行


选择 Build->Execute MyFile.ext 选项可以启动程序的执行。此外，也可以通过点击  按钮或通过快捷键 Ctrl+F5 启动程序的运行。开始运行后，将弹出一个 DOS 窗口，显示程序的运行结果，如图 0.15 所示。




图 0.15 程序运行结果

## 四、C 程序的调试

调试是为了发现程序中的错误，包括语法错误和逻辑错误。其中，语法错误能够在编译的过程中发现并修改；而逻辑错误往往无法直观地被发现，即程序通常能够被成功地编译和连接甚至执行，然而其执行结果却与预期结果不一致。逻辑错误的调试是比较困难的，因此，一般的程序开发环境都会提供完整的程序调试工具。本节将主要介绍如何使用 Visual C++ 6.0 中的调试工具发现程序中的逻辑错误。

### 1、设置断点

当需要调试程序的时候，可以首先大致判断出程序中可能从哪条语句开始出现问题，并

将光标移动到该语句行，点击按钮，则在语句行左侧出现一个红点，称为断点，如图 0.16 所示。此后，程序调试运行过程中遇到断点时，将自动暂停执行，进入调试状态。

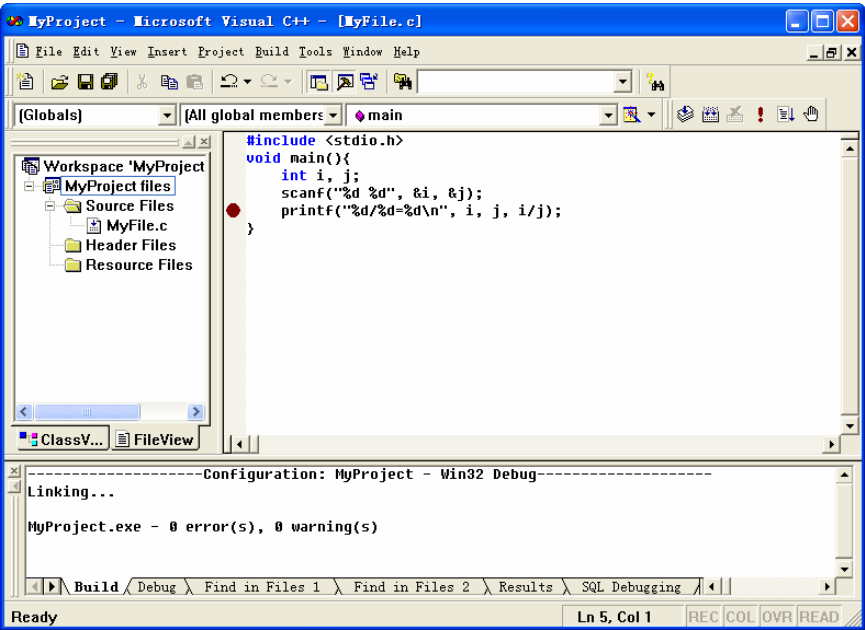



图 0.16 设置断点

## 2、进入调试界面

设置好断点之后，点击按钮，以 debug 的方式启动程序的运行。当运行到断点所在的语句时，将直接进入 debug 状态，如图 0.17 所示。代码行左侧的黄色小箭头表示了程序的当前执行位置。Debug 工具栏中是常用的调试工具按钮，表 0.1 中列举了常用按钮的功能。屏幕下方左右两个窗口则分别是自动变量框和观察框，可以看到变量的当前值，作为判断程序是否出错的参考。其中，观察框支持对各种变量值的观察，在 Name 列输入变量的名字，则在 Value 列就会显示出该变量的当前值。

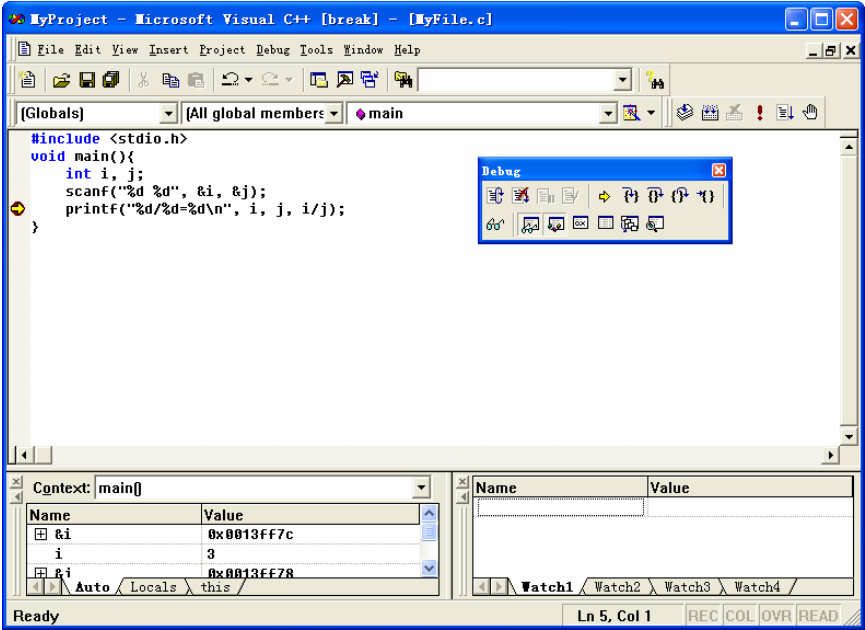


图 0.17 Debug 界面

表 0.1 Debug 工具条常用按钮说明

按 钮	功 能	按 钮	功 能
	重新开始 Debug		停止/退出 Debug
	跟踪到函数中		逐个函数调试
	从跟踪的函数中跳出		运行到光标处

此后，使用上述调试工具则可以跟踪程序的执行过程，并通过观察变量的内容变化即时发现程序的错误所在。

# 第一章 学生信息管理

本章要求设计一个学生信息管理程序，要求实现学生基本信息的登记、学生考试成绩的登记和奖学金的评定等功能；使用结构体存储学生的基本信息及考试情况；使用数组来实现对学生信息及其考试成绩的添加、删除、查询、修改等操作；使用文件来保存数据，下次运行时可以从文件自动读取数据。

## 一、实践目的

- 1、掌握结构体数组的基本工作原理和处理方式。
- 2、比较结构体数组与链表的使用方法。
- 3、加深理解多文档编程的思想，掌握对各个源文件进行处理的方式。
- 4、会使用 C 语言对文件进行读取、修改等操作。
- 5、学会熟练使用各种结构来完成程序。

## 二、基本要求

- 1、本程序要求实现对学生信息的查找、添加、删除、修改、浏览、保存、从文件读取、查看奖学金信息八个功能，每个功能模块均能实现随时从模块中退出，而且可以选择不同的方式实现所需功能，从而完成一个学生管理系统所需功能。
- 2、要使用结构体中嵌有结构体来实现对学生信息的存储。
- 3、使用链表来实现对学生信息的查找、添加、删除、修改、浏览等操作。
- 4、使用文件完成数据的存储与读取，要求每次运行某个功能模块时将数据读入结构体中，并给用户提供保存选项，可以将结构体中的数据保存在文件中。
- 5、系统制作完成后应实现类似图 1.1 所示界面。

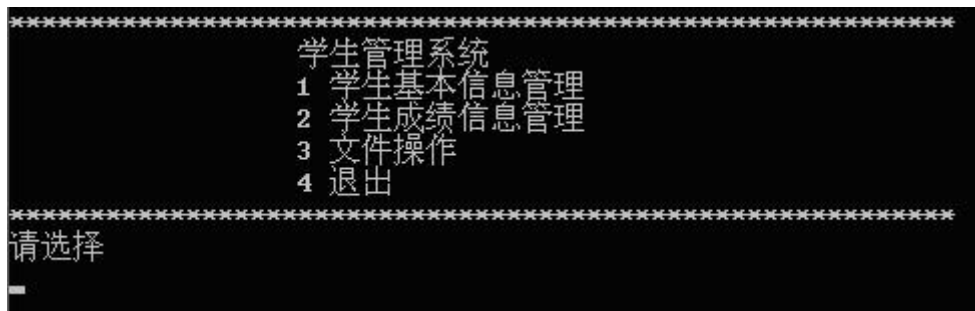


图 1.1 程序运行界面

## 三、算法分析

### 1、数据结构

程序需要结构体中嵌有结构体的结构体来实现对学生信息的存储，结构体设计参考如下：

(1) 定义学生的基本信息

```
typedef struct studentMessage{  
    char name[15]; //姓名  
    int id; //学号  
    char sex[7]; //性别
```

```

char collage[20];//学院
int Class;//班级
struct studentMessage *next;//定义链表
} MESSAGE;

```

(2) 定义学生的成绩信息

```

typedef struct studentMark{
    int id;//学号
    char name[15];//姓名
    struct subject{
        int subid;
        char subname[11];
        float score;
    } sub[length];//保存学生各科成绩
    double total;//总分
    struct studentMark *next;
} MARK;

```

## 2、函数定义

程序由 main.c、stumessage.c、stufile.c、scholarship.c 一共 4 个源文件构成，函数说明如表 1.1 所示。

表 1.1 文件及函数说明

源文件	函数原型	功能说明
include.h	头文件定义	
	宏定义	
	结构体定义	
	自定义函数声明	
main.c	void main()	程序的入口
	void mainmenu(MESSAGE *head , Mark * top)	处理菜单的选择
	int menu(void)	实现对菜单的选择
stuMessage.c	void searchmenu(MESSAGE *head)	查找学生基本信息的菜单
	MESSAGE *addstudent(MESSAGE *head)	添加学生的基本信息
	MESSAGE *deletestuone(MESSAGE *head)	删除某个学生的基本信息
	MESSAGE *deletestuall(MESSAGE *head)	删除所有学生的基本信息
	void searchstuone(MESSAGE * )	查找某个学生的基本信息
	MESSAGE *deletestuclass(MESSAGE*head)	删除某个班的学生的基本信息
	void scanstuAll(MESSAGE *head)	浏览所有学生的基本信息
	MESSAGE *deletemenu(MESSAGE*head)	删除学生的基本信息的菜单
	MESSAGE *changestu(MESSAGE *head)	修改学生的基本信息的主控函数
	MESSAGE *change(MESSAGE *head)	修改学生基本信息的具体函数
	void searchstuclass(MESSAGE * )	查找某个班级学生的基本信息
stuMark.c	void searchMarkmenu(MARK *top)	查找学生成绩信息的菜单
	MARK *addstudentMark(MARK *top)	增加学生的成绩信息
	void scanstuMarkAll(MARK *top)	查看所有学生的成绩信息
	void searchstuMarkone(MARK * p)	查找某一学生的成绩信息
	MARK *changestuMark(MARK *top)	修改学生成绩信息的菜单
	MARK * changeMark(MARK *p)	修改学生的成绩信息
	MARK *deleteMarkmenu(MARK *top)	删除学生成绩信息的菜单

	MARK *deletestuMarkone(MARK *top)	删除某一学生的成绩信息
	MARK *deletestuMarkall(MARK *top)	删除所有学生的成绩信息
	void printscholarship(MARK *top)	打印学生的奖学金信息
stufile.c	MESSAGE *readfilestu(MESSAGE *)	从文件中读取学生的基本信息
	MESSAGE *readfilestu(MESSAGE *head)	从文件中读取学生的成绩信息
	void savefilestu(MESSAGE *head)	保存学生的基本信息到文件中
	void savefilestu(MESSAGE *head)	保存学生的成绩信息到文件中

### 3、处理过程

#### (1) 用户选择功能的实现

通过调用 menu()函数来实现用户所需要使用功能的选择的选项，再调用 mainmenu()来实现用户所需功能的函数，完成所需的功能。

#### (2) 学生基本信息的管理

调用 addstudent()实现增加学生的信息，调用 searchmenu()实现浏览学生的信息的菜单，以方便用户使用该功能；调用 deletemenu()实现删除学生信息菜单的功能；调用 changestu()完成修改学生信息的功能。

#### (3) 学生成绩信息管理的实现

实现学生成绩信息的管理与实现学生基本信息管理过程基本相同，在这里就不一一列举。

#### (4) 学生信息文件操作的管理实现

通过调用 savefilestu()和 savefilestuMark()实现将学生的信息保存到文件中，调用 readfilestu()和 readfilestuMark()将从文件中读取学生的信息。

下面仅以将学生的基本信息保存到文件中的 void savefilestu(MESSAGE \*head)函数为例介绍基本处理过程：

1) 声明文件指针和链表指针，并且判断是否为其分配内存，如果申请内存失败，退出本系统，否则，继续运行本系统。

```
FILE *stu;
MESSAGE *p;
p=MES;
if(p==NULL){
    printf("申请内存失败!\n");
    exit(1);
}
```

2) 设置打开文件的方式，

```
stu=fopen("STUDENT.txt","w");break;
```

一般用“a+”方式将信息保存到文件中，但这里选择了以“w”的方式保存信息，两者的区别在于“a+”以添加到文件的结尾，而“w”则是重写文件的信息，至于用何种方式打开文件没有规定，最终要根据具体情况而定。

3) 运用头指针的循环将所有的信息保存到文件中，并且将保存的信息打印在屏幕上(可选)，主要用 fprintf 与 fscanf 实现：

```
do{
    printf("姓名\t学号\t性别\t学院\t班级\n");//读入文件并且打印在屏幕上
```



```

fprintf(stu,"%s\\t%d\\t%s\\t%s\\t%d\\t",p->name,p->id,p->sex,p->collage,p->Class);
printf("%s\\t%d\\t%s\\t%s\\t%d\\n",p->name,p->id,p->sex,p->collage,p->Class);
    for(i=0;i<10;i++){
        printf("%s\\t",subname[i]);
    }
    p=p->next;//循环遍历链表
}while(p!=NULL);

```

#### 四、参考代码

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
//宏定义
#define length 10
#define MES (struct studentMessage *)malloc(sizeof(struct studentMessage));
#define MAR (struct studentMark *)malloc(sizeof(struct studentMark));
char subname[][9]={"高数","高代","C 语言","英语","工概","心健","思修","德育","体育","奖励"};
//定义结构体
typedef struct studentMessage{
    char name[15];
    int id;
    char sex[7];
    char collage[20];
    int Class;
    struct studentMessage *next;
} MESSAGE;
typedef struct studentMark{
    int id;
    char name[15];
    struct subject{
        int subid;
        char subname[11];
        float score;
    } sub[length];
    double total;
    struct studentMark *next;
} MARK;
void mainmenu(MESSAGE *,MARK *);
void searchmenu(MESSAGE *);
int menu(void);
MESSAGE *addstudent(MESSAGE *);
void scanstuAll(MESSAGE *);
void searchstuclass(MESSAGE *);
MESSAGE *changestu(MESSAGE*);
MESSAGE *change(MESSAGE*);
MESSAGE *deletemenu(MESSAGE *);
void searchstuone(MESSAGE *);
MESSAGE *deletestuone(MESSAGE *);
MESSAGE *deletestuall(MESSAGE *);
MESSAGE *deletestuclass(MESSAGE *);
MARK *addstudentMark(MARK *);
void scanstuMarkAll(MARK *);
void searchMarkmenu(MARK *);
void searchstuMarkone(MARK *);

```

```

MARK *deleteMarkmenu(MARK *);
MARK *deletestuMarkall(MARK *);
MARK *deletestuMarkone(MARK *);
MARK * changeMark(MARK *);
MARK *changestuMark(MARK *);
void printscholarship(MARK *);
MESSAGE *readfilestu(MESSAGE *);
MARK *readfilestuMark(MARK *);
void savefilestuMark(MARK *);
void savefilestu(MESSAGE *);
MARK *readfilestuMark(MARK *);
MESSAGE *readfilestu(MESSAGE *);

//主函数
main(){
    MESSAGE *head=NULL;
    MARK *top=NULL;
    mainmenu(head,top);
}
//控制菜单选项
void mainmenu(MESSAGE *head,MARK *top){
    int getchoose,choose,choose1;
    while(1){
        getchoose=menu();
        switch (getchoose){
            case 1:
                system("cls");
                printf("          学生基本信息管理\n");
                printf("          1 查找信息\n");
                printf("          2 添加信息\n");
                printf("          3 删除信息\n");
                printf("          4 修改信息\n");
                printf("          5 返回\n");
                fflush(stdin);
                printf("请选择\n");
                scanf("%d",&choose);
                if(choose==1){
                    searchmenu(head);
                }
                else if(choose==2){
                    head=addstudent(head);
                }
                else if(choose==3){
                    head=deletemenu(head);
                }
                else if(choose==4){
                    head=changestu(head);
                }
                else if(choose==5){
                    printf("请按任意键返回\n");
                    getch();
                }
                else{
                    printf("您输入的错误\n");
                }
            break;
            case 2:
                system("cls");

```

```

printf("                                学生成绩信息管理\n");
printf("                                1 查找成绩信息\n");
printf("                                2 添加成绩信息\n");
printf("                                3 删除成绩信息\n");
printf("                                4 修改成绩信息\n");
printf("                                5 奖学金信息\n");
printf("                                6 返回\n");
fflush(stdin);
printf("请选择\n");
scanf("%d",&choose);
if(choose==1){
    searchMarkmenu(top);
}
else if(choose==2){
    top=addstudentMark(top);
}
else if(choose==3){
    top=deleteMarkmenu(top);
}
else if(choose==4){
    top=changestuMark(top);
}
else if(choose==5){
    printscholarship(top);
}
else if(choose==6){
    printf("请按任意键返回\n");
    getch();
}
else{
    printf("您输入的错误\n");
}
break;
case 3:
    system("cls");
    printf("    文件管理\n");
    printf("    1 从文件中读取信息\n");
    printf("    2 保存学生信息到文件中\n");
    printf("    3 返回\n");
    fflush(stdin);
    printf("请选择\n");
    scanf("%d",&choose);
    if(choose==1){
        printf("    1 读取学生基本信息\n");
        printf("    2 读取学生成绩信息\n");
        fflush(stdin);
        printf("请选择\n");
        scanf("%d",&choose1);
        if(choose1==1){
            head=readfilestu(head);
        }
        if(choose1==2){
            top=readfilestuMark(top);
        }
        if(choose1!=1&&choose1!=2){
            printf("你输入的错误\n");
            printf("请按任意键返回\n");
            getch();
        }
    }

```

```

    }
}
if(choose==2){
    printf("    1  保存学生基本信息\n");
    printf("    2  保存学生成绩信息\n");
    fflush(stdin);
    printf("请选择\n");
    scanf("%d",&choose1);
    if(choose1==1){
        savefilestu(head);
    }
    if(choose1==2){
        savefilestuMark(top);
    }
    if(choose1!=1&&choose1!=2){
        printf("你输入的错误\n");
        printf("请按任意键返回\n");
        getch();
    }
}
if(choose==3){
    printf("请按任意键返回\n");
    getch();
}
break;
case 4:
    exit(0);break;
}
}
}

```

```

void searchmenu(MESSAGE *head){
    int choose;
    printf("1  个人信息\n");
    printf("2  班级信息\n");
    printf("3  全部信息\n");
    printf("请选择\n");
    fflush(stdin);
    scanf("%d",&choose);
    switch(choose){
        case 1:searchstuone(head);break;
        case 2:searchstuclass(head);break;
        case 3:scanstuAll(head);break;
        default :printf("你输入胡错误!\n");break;
    }
}
//菜单选项
int menu(void){
    int choose;
    do{
        system("cls");
        printf("*****\n");
        printf("                学生管理系统\n");
        printf("1  学生基本信息管理\n");
        printf("2  学生成绩信息管理\n");
        printf("3  文件操作\n");
        printf("4  退出\n");
        printf("*****\n");
    }
}

```

```

        fflush(stdin);
        printf("请选择\n");
        scanf("%d",&choose);
        if(choose!=1&&choose!=2&&choose!=3&&choose!=4){
            printf("你输入的错误，请重新输入!\n");
        }
    }while(choose!=1&&choose!=2&&choose!=3&&choose!=4);
    return choose;
}
//添加学生信息
MESSAGE *addstudent(MESSAGE *head){
    int num;
    MESSAGE *p1, *p2;
    do{
        fflush(stdin);
        printf("请输入一个学生的学号:(以输入 0 结束!)\n ");
        scanf("%d", &num);
    }while(num<0);
    while(num!=0){//输入信息
        p1=MES;
        fflush(stdin);
        printf("请输入该学生的姓名:\n");
        scanf("%s",p1->name);
        do{
            fflush(stdin);
            printf("请输入该学生的性别!(male/female)\n");
            scanf("%s",p1->sex);
        }while(strcmp(p1->sex,"male")!=0&&strcmp(p1->sex,"female")!=0);
        fflush(stdin);
        printf("请输入该学生所在学院\n");
        scanf("%s",p1->collage);
        do{
            fflush(stdin);
            printf("请输入该学生所在班级\n");
            scanf("%d",&p1->Class);
        }while(p1->Class<0);
        p1->id=num;
        p1->next=NULL;
        if(head!=NULL){
            p2=head;
            p1->next=p2;
            head=p1;
        }
        else
            head=p1;
        do{
            fflush(stdin);
            printf("请输入一个学生的学号:(以输入 0 结束!)\n ");
            scanf("%d", &num);
        }while(num<0);
    }
    return head;
}
//浏览学生信息
void scanstuAll(MESSAGE *head){
    int count=0;
    MESSAGE *p=head;
    while(p!=NULL){

```

```

        printf("姓名\t 学号\t 性别\t 学院\t 班级\t\n");
printf("%s\t%d\t%s\t%s\t%d\t\n",p->name,p->id,p->sex,p->collage,p->Class);
        p=p->next;
        count++;
    }
    printf("总共有%d 条学生信息!",count);
    printf("请按任意键返回主菜单!\n");
    getch();
}
//查找学生的基本信息
void searchstuone(MESSAGE * p){
    char name[12];
    int id;
    int gole=0;
    int choose;
    do{
        fflush(stdin);
        printf("请选择!(1 姓名 2 学号 3 返回)\n");
        scanf("%d",&choose);
        switch(choose){
            case 1:
                fflush(stdin);
                printf("请输入要查找的名字\n");
                gets(name);
                while(p!=NULL){
                    if(!strcmp(name,p->name)){
                        printf("姓名\t 学号\t 性别\t 学院\t 班级\t\n");
printf("%s\t%d\t%s\t%s\t%d\t\n",p->name,p->id,p->sex,p->collage,p->Class);
                        gole++;
                    }
                    printf("\n");
                    p=p->next;
                }
                if(gole==0){
                    printf("没有您要找的信息!\n");
                }break;
            case 2:
                fflush(stdin);
                printf("请输入要查找的学号\n");
                scanf("%d",&id);
                while(p!=NULL){
                    if(id=p->id){
                        printf("姓名\t 学号\t 性别\t 学院\t 班级\t\n");
printf("%s\t%d\t%s\t%s\t%d\t\n",p->name,p->id,p->sex,p->collage,p->Class);
                        gole++;
                    }
                    printf("\n");
                    p=p->next;
                }
                if(gole==0){
                    printf("没有您要找的信息!\n");
                }break;
            case 3:printf("请按任意键返回!\n");getch();break;
            default :printf("您输入的的错误!\n 请重新选择\n");break;
        }
    }while(choose!=1&&choose!=2&&choose!=3);
    getch();
}

```

```

void searchstuclass(MESSAGE * p){
    int Class;
    int gole=0;
    fflush(stdin);
    printf("请输入要查找的班级\n");
    scanf("%d",&Class);
    while(p!=NULL){
        if(Class=p->Class){
            printf("姓名\t 学号\t 性别\t 学院\t 班级\n");
            printf("%s\t%d\t%s\t%s\t%d\n",p->name,p->id,p->sex,p->collage,p->Class);
            gole++;
        }
        printf("\n");
        p=p->next;
    }
    if(gole==0){
        printf("该班级中没有任何信息!\n");
    }
    else{
        printf("该班级中总共有%d 信息!\n",gole);
    }
    getch();
}
//修改学生的信息
MESSAGE *changestu(MESSAGE*head){
    char name[12];
    int id;
    int gole=0;
    int choose;
    MESSAGE *p;
    p=MES;
    if(p==NULL){
        printf("分配内存失败!");
        exit(1);
    }
    if(head!=NULL){
        do{
            fflush(stdin);
            printf("请选择(1 姓名 2 学号 3 返回)\n");
            scanf("%d",&choose);
            p=head;
            if(choose==1){
                fflush(stdin);
                printf("请输入要修改的信息的名字\n");
                gets(name);
                while(p!=NULL){
                    if(strcmp(name,p->name)==0){
                        p=change(p);
                        gole++;
                    }
                    p=p->next;
                }
                if(gole==0){
                    printf("没有您要修改的信息!\n");
                }
            }
            else if(choose==2){

```

```

        fflush(stdin);
        printf("请输入要修改的信息的学号\n");
        scanf("%d",&id);
        while(p!=NULL){
            if(id==p->id){
                p=change(p);
                gole++;
            }
            p=p->next;
        }
        if(gole==0){
            printf("没有您要修改的信息!\n");
        }
    }
    else if(choose==3){
        printf("请按任意键返回主菜单!\n");getch();break;
    }
    else{
        printf("您输入的错误，请重新输入!\n");
    }
} while(choose!=1&&choose!=2&&choose!=3);
}
else{
    printf("没有信息，请先存入信息!\n");
}
printf("请按任意键返回主菜单!\n");getch();
return head;
}

//分别修改学生的信息
MESSAGE * change(MESSAGE *p){
    int choose1;
    printf("姓名\t学号\t性别\t学院\t班级\t\n");
    printf("%s\t%d\t%s\t%s\t%d\t",p->name,p->id,p->sex,p->collage,p->Class);
    do{
        fflush(stdin);
        printf("请选择(1 姓名 2 学号 3 性别 4 班级 5 学院 6 返回)\n");
        //选择修改的信息
        scanf("%d",&choose1);
        switch(choose1){
            case 1:
                fflush(stdin);
                printf("请输入要修改的名字!\n");
                gets(p->name);
                printf("已经成功修改!\n");
                break;
            case 2:
                fflush(stdin);
                printf("请输入要修改的学号!\n");
                scanf("%d",&p->id);
                printf("已经成功修改!\n");
                break;
            case 3:
                fflush(stdin);
                printf("请输入要修改的性别!\n");
                gets(p->sex);
                printf("已经成功修改!\n");
                break;

```



```

        case 4:
            fflush(stdin);
            printf("请输入要修改的班级!\n");
            scanf("%d",&p->Class);
            printf("已经成功修改!\n");

            break;
        case 5:
            fflush(stdin);
            printf("请输入要修改的学院!\n");
            gets(p->collage);
            printf("已经成功修改!\n");
        case 6:break;
        default :printf("您输入的错误!\n 请重新选择\n");break;
    }
    while(choose1!=1&&choose1!=2&&choose1!=3&&choose1!=4&&choose1!=5&&choose1!=6);

    getch();
    return p;
}

```

```

MESSAGE *deletemenu(MESSAGE *head){
    int choose;
    printf("1 删除个人信息\n");
    printf("2 删除班级信息\n");
    printf("3 删除全部信息\n");
    fflush(stdin);
    printf("请选择\n");
    scanf("%d",&choose);
    if(choose==1){
        head=deletestuone(head);
    }
    if(choose==2){
        head=deletestuclass(head);
    }
    if(choose==3){
        head=deletestuall(head);
    }
    if(choose!=1&&choose!=2&&choose!=3){
        printf("你输入的错误!\n");
    }
    return head;
}

```

```

MESSAGE *deletestuclass(MESSAGE *head){
    int Class,gole=0;
    MESSAGE *p1,*p2,*p3=head;
    fflush(stdin);
    printf("请输入你要删除的班级!\n");
    scanf("%d",&Class);
    while(p3!=NULL){
        p1=head;
        p2=head;
        while((p1->next!=NULL)&&(p1->Class!=Class)){
            p2=p1;
            p1=p1->next;
        }
        if(p1->Class==Class){

```

```

        printf("姓名\t 学号\t 性别\t 学院\t 班级\t");
printf("%s\t%d\t%s\t%s\t%d\t",p1->name,p1->id,p1->sex,p1->collage,p1->Class);
        if(p1==head)
            head=p1->next;
        else
            p2->next=p1->next;
        free(p1);
        gole++;
    }else
        printf("该班没有任何信息\n");
    p3=p3->next;
}
if(gole!=0)
    printf("共删除该班%d 条信息",gole);
getch();
return head;
}

```

```

MESSAGE *deletestual(MESSAGE *head){
    int count=0;
    while(head!=NULL){
        count++;
        head=head->next;
    }
    printf("总共删除%d 条信息",count);
    return head;
}

```

//删除个人学生的信息

```

MESSAGE *deletestuone(MESSAGE *head){
    char name[12];
    int id;
    int gole=0;
    int choose;
    char m;
    MESSAGE *p1,*p2,*find;
    p1=MES;
    p2=MES;
    find=MES;
    if(p1==NULL){//分配内存
        printf("分配内存失败!");
        exit(1);
    }
    if(p2==NULL){
        printf("分配内存失败!");
        exit(1);
    }
    if(find==NULL){
        printf("分配内存失败!");
        exit(1);
    }
    if(head!=NULL){
        p1=head;
        p2=head;
        do{
            fflush(stdin);
            printf("请选择!(1 姓名 2 学号 3 返回)\n");
            scanf("%d",&choose);

```

```

switch(choose){
    case 1:
        find=head;
        fflush(stdin);
        printf("请输入要删除信息的名字\n");
        gets(name);
        while(find!=NULL){
            if(!strcmp(name,find->name)){
                printf("姓名\t学号\t性别\t学院\t班级\t");
                printf("%s\t%d\t%s\t%s\t%d\t",find->name,find->id,find->sex,find->collage,find->Class);
                gole++;
            }
            printf("\n");
            find=find->next;
        }
        if(gole==0){
            printf("没有您要删除的信息!\n");
        }
        else{
            fflush(stdin);
            printf("是否要删除以上信息(y/n)\n");
            m=getchar();
            if(m=='y'||m=='Y'){
                while((p1->next!=NULL)&&(strcmp(name,p1->name)!=0)){
                    p2=p1;p1=p1->next;
                }
                if(strcmp(name,p1->name)==0){
                    if(p1==head)
                        head=p1->next;
                    else
                        p2->next=p1->next;
                    free(p1);
                }
                printf("成功删除以上信息!\n");
            }
        }
        printf("请按任意键返回主菜单!\n");getch();
        break;
    case 2:
        fflush(stdin);
        printf("请输入要删除信息的学号\n");
        scanf("%d",&id);
        find=head;
        while(find!=NULL){
            if(id==find->id){
                printf("姓名\t学号\t性别\t学院\t班级\t");
                printf("%s\t%d\t%s\t%s\t%d\t",find->name,find->id,find->sex,find->collage,find->Class);
                gole++;
            }
            printf("\n");
            find=find->next;
        }
        if(gole==0){
            printf("没有您要删除的信息!\n");
        }
        else{
            fflush(stdin);
            printf("是否要删除以上信息(y/n)\n");
            m=getchar();

```

```

        if(m=='y'||m=='Y'){
            while((p1->next!=NULL)&&(id!=p1->id)){
                p2=p1;p1=p1->next;
            }
            if(id==p1->id){
                if(p1==head)
                    head=p1->next;
                else
                    p2->next=p1->next;
                free(p1);
            }
            printf("成功删除以上信息!\n");
        }
    }
    printf("请按任意键返回主菜单!\n");getch();
    break;
case 3: printf("请按任意键返回!\n");getch();break;
default :printf("您输入的错误的!\n 请重新选择\n");break;
}
} while(choose!=1&&choose!=2&&choose!=3);
}
else{
    printf("没有信息，请先存入信息!\n");
    printf("请按任意键返回主菜单!\n");
    getch();
}
}
return head;
}

void searchMarkmenu(MARK *top){
    int choose;
    printf("1 个人成绩\n");
    printf("2 全部学生成绩\n");
    printf("请选择\n");
    fflush(stdin);
    scanf("%d",&choose);
    switch(choose){
        case 1:searchstuMarkone(top);break;
        case 2:scanstuMarkAll(top);break;
        default :printf("你输入胡错误!\n");break;
    }
}

//添加学生信息
MARK *addstudentMark(MARK *top){
    int num,i,j,ID,gole;
    MARK *p1=top, *p2=top,*p;
    do{
        fflush(stdin);
        printf("请输入一个学生的学号:(以输入 0 结束!)\n ");
        scanf("%d", &num);
    }while(num<0);
    while(num!=0){//输入信息
        gole=0;p1=top;p2=top;
        p=MAR;
        p->total=0;
        fflush(stdin);
        printf("请输入该学生的姓名:\n");
        scanf("%s",p->name);
    }
}

```

```

        for(i=0;i<length;i++){
            printf("%d %s;",i+1,subname[i]);
        }
        printf("\n");
        fflush(stdin);
        printf("请选择你所选修的课程(输入-1 结束选择)\n");
        for(i=0;i<length;i++){
            scanf("%d",&ID);
            p->sub[i].subid=ID;
            if(ID==-1){
                break;
            }
            gole++;
        }
        for(i=0;i<gole;i++){
            j=p->sub[i].subid;
            if(j!=-1){
                fflush(stdin);
                printf("请输入该学生%s的成绩:\n",subname[j-1]);
                scanf("%f",&p->sub[j-1].score);
                p->total+=((double)(p->sub[j-1].score)/gole);
            }
            else
                break;
        }
        p->id=num;
        p->next=NULL;
        if(!top){
            top=p;
        }
        else {
            //找到插入的位置
            while(p1){
                if(p1->total>=p->total){
                    p2=p1;
                    p1=p1->next;
                }
                else
                    break;
            }
            p->next=p1;
            p2->next=p;
        }
        do{
            fflush(stdin);
            printf("请输入一个学生的学号:(以输入 0 结束!)\n ");
            scanf("%d", &num);
        }while(num<0);
    }
    return top;
}
//浏览学生信息
void scanstuMarkAll(MARK *top){
    int count=0,i,j,k=0;
    MARK *p=top;
    while(p!=NULL){
        printf("姓名\t 学号\n");
        printf("%s\t%d\n",p->name,p->id);
    }
}

```

```

        for(i=0;i<length;i++){
            j=p->sub[i].subid;
            if(j==-1)
                break;
            printf("%s\t",subname[j-1]);
            printf("%.2f\n",p->sub[j-1].score);
        }
        printf("总分\t");
        printf("%.2lf\n",p->total);
        p=p->next;
        count++;
    }
    printf("总共有%d 条学生信息!\n",count);
    printf("请按任意键返回主菜单!\n");
    getch();
}
//查找学生的基本信息
void searchstuMarkone(MARK * p){
    char name[12];
    int id,i,j;
    int gole=0;
    int choose;
    do{
        fflush(stdin);
        printf("请选择!(1 姓名 2 学号 3 返回)\n");
        scanf("%d",&choose);
        switch(choose){
            case 1:
                fflush(stdin);
                printf("请输入要查找的名字\n");
                gets(name);
                while(p!=NULL){
                    if(!strcmp(name,p->name)){
                        printf("姓名\t学号\n");
                        printf("%s\t%d\n",p->name,p->id);
                        for(i=0;i<length;i++){
                            j=p->sub[i].subid;
                            if(j==-1)
                                break;
                            printf("%s\n",subname[j-1]);
                            printf("%.2f\n",p->sub[j-1].score);
                            printf("\n");
                        }
                        printf("总分\n");
                        printf("%.2lf\n",p->total);
                        gole++;
                    }
                    printf("\n");
                    p=p->next;
                }
            case 2:
                fflush(stdin);
                printf("请输入要查找的学号\n");
                scanf("%d",&id);
                while(p!=NULL){

```

```

        if(id=p->id){
            printf("姓名\t 学号\n");
            printf("%s\t%d\n",p->name,p->id);
            for(i=0;i<length;i++){
                j=p->sub[i].subid;
                if(j==-1)
                    break;
                printf("%s\n",subname[j-1]);
                printf("%.2f\n",p->sub[j-1].score);
                printf("\n");
            }
            printf("总分\n");
            printf("%.2lf\n",p->total);
            gole++;
        }
        printf("\n");
        p=p->next;
    }
    if(gole==0){
        printf("没有您要找的信息!\n");
    }break;
case 3:printf("请按任意键返回!\n");getch();break;
default :printf("您输入的错误的!\n 请重新选择\n");break;
    }
}while(choose!=1&&choose!=2&&choose!=3);
printf("请按任意键返回!\n");
getch();
}
//修改学生的信息
MARK *changestuMark(MARK *top){
    char name[12];
    int id;
    int gole=0;
    int choose;
    MARK *p;
    p=MAR;
    if(p==NULL){
        printf("分配内存失败!");
        exit(1);
    }
    if(top!=NULL){
        do{
            fflush(stdin);
            printf("请选择(1 姓名 2 学号 3 返回)\n");
            scanf("%d",&choose);
            p=top;
            if(choose==1){
                fflush(stdin);
                printf("请输入要修改的信息的名字\n");
                gets(name);
                while(p!=NULL){
                    if(strcmp(name,p->name)==0){
                        p=changeMark(p);
                        gole++;
                    }
                    p=p->next;
                }
            }
            if(gole==0){

```

```

        printf("没有您要修改的信息!\n");
    }
}
else if(choose==2){
    fflush(stdin);
    printf("请输入要修改的信息的学号\n");
    scanf("%d",&id);
    while(p!=NULL){
        if(id==p->id){
            p=changeMark(p);
            gole++;
        }
        p=p->next;
    }
    if(gole==0){
        printf("没有您要修改的信息!\n");
    }
}
else if(choose==3){
    printf("请按任意键返回主菜单!\n");getch();break;
}
else {
    printf("您输入的错误，请重新输入!\n");
}
} while(choose!=1&&choose!=2&&choose!=3);
}
else{
    printf("没有信息，请先存入信息!\n");
}
printf("请按任意键返回主菜单!\n");getch();
return top;
}
//分别修改学生的信息
MARK * changeMark(MARK *p){
    int choose1,i,j,k=3,m,gole=0;
    float change=0;
    printf("姓名\t 学号\n");
    printf("%s\t%d\n",p->name,p->id);
    for(i=0;i<length;i++){
        j=p->sub[i].subid;
        if(j== -1)
            break;
        gole++;
        printf("%s\n",subname[j-1]);
        printf("%.2f\n",p->sub[j-1].score);
        printf("\n");
    }
    printf("总分\n");
    printf("%.2lf\n",p->total);
    do{
        fflush(stdin);
        printf("请选择(1 姓名 2 学号 );//选择修改的信息
        for(i=0;i<length;i++){
            j=p->sub[i].subid;
            if(j== -1)
                break;
            printf("%d   %s",i+3,subname[j-1]);
        }
    }

```



```

        printf("\n");
scanf("%d",&choose1);
    if(choose1==1){
        fflush(stdin);
        printf("请输入要修改的名字!\n");
        gets(p->name);
        printf("已经成功修改!\n");
    }
    if(choose1==2){
        fflush(stdin);
        printf("请输入要修改的学号!\n");
        scanf("%d",&p->id);
        printf("已经成功修改!\n");
    }
    while(1){
        if(choose1==k){
            fflush(stdin);
            m=p->sub[k-3].subid;
            printf("请输入要修改的%s 的分数!\n",subname[m-1]);
            change=p->sub[m-1].score;
            scanf("%.3f",&p->sub[m-1].score);
            p->total+=(int)(p->sub[m-1].score-change)/gole;
            printf("已经成功修改!\n");
            break;
        }
        k++;
    }
} while(choose1!=1&&choose1!=2&&choose1!=3&&choose1!=4&&choose1!=5&&choose1!=6);

    getch();
    return p;
}

```

```

MARK *deleteMarkmenu(MARK *top){
    int choose;
    printf("1 删除个人信息\n");
    printf("2 删除全部信息\n");
    fflush(stdin);
    printf("请选择\n");
    scanf("%d",&choose);
    if(choose==1){
        top=deletestuMarkone(top);
    }
    if(choose==2){
        top=deletestuMarkall(top);
    }
    if(choose!=1&&choose!=2){
        printf("你输入的错误!\n");
    }
    return top;
}
//删除个人学生的信息
MARK *deletestuMarkone(MARK *top){
    char name[12];
    int id,i,j;
    int gole=0;
    int choose;
    char m;

```

```

MARK *p1,*p2,*find;
p1=MAR;
p2=MAR;
find=MAR;
if(p1==NULL){//分配内存
    printf("分配内存失败!");
    exit(1);
}
if(p2==NULL){
    printf("分配内存失败!");
    exit(1);
}
if(find==NULL){
    printf("分配内存失败!");
    exit(1);
}
if(top!=NULL){
    p1=top;
    p2=top;
    do{
        fflush(stdin);
        printf("请选择!(1 姓名 2 学号 3 返回)\n");
        scanf("%d",&choose);
        switch(choose){
            case 1:
                find=top;
                fflush(stdin);
                printf("请输入要删除信息的名字\n");
                gets(name);
                while(find!=NULL){
                    if(!strcmp(name,find->name)){
                        printf("姓名\t学号\n");
                        printf("%s\t%d\n",find->name,find->id);
                        for(i=0;i<length;i++){
                            j=find->sub[i].subid;
                            if(j==-1)
                                break;
                            printf("%s\n",subname[j-1]);
                            printf("%.2f\n",find->sub[j-1].score);
                            printf("\n");
                        }
                        printf("总分\n");
                        printf("%.2lf\n",find->total);
                        gole++;
                    }
                    printf("\n");
                    find=find->next;
                }
            if(gole==0){
                printf("没有您要删除的信息!\n");
            }
            else{
                fflush(stdin);
                printf("是否要删除以上信息(y/n)\n");
                m=getchar();
                if(m=='y'||m=='Y'){
                    while((p1->next!=NULL)&&(strcmp(name,p1->name)!=0)){
                        p2=p1;p1=p1->next;
                    }
                }
            }
        }
    }while(p1!=NULL);
}

```

```

        }
        if(strcmp(name,p1->name)==0){
            if(p1==top)
                top=p1->next;
            else
                p2->next=p1->next;
            free(p1);
        }
        printf("成功删除以上信息!\n");
    }
} printf("请按任意键返回主菜单!\n");getch();
break;
case 2:
    fflush(stdin);
    printf("请输入要删除信息的学号\n");
    scanf("%d",&id);
    find=top;
    while(find!=NULL){
        if(id==find->id){
            printf("姓名\t学号\n");
            printf("%s\t%d\n",find->name,find->id);
            for(i=0;i<length;i++){
                j=find->sub[i].subid;
                if(j==-1)
                    break;
                printf("%s\n",subname[j-1]);
                printf("%.2f\n",find->sub[j-1].score);
                printf("\n");
            }

            printf("总分\n");
            printf("%.2lf\n",find->total);
            gole++;
        }
        printf("\n");
        find=find->next;
    }
    if(gole==0){
        printf("没有您要删除的信息!\n");
    }
    else{
        fflush(stdin);
        printf("是否要删除以上信息(y/n)\n");
        m=getchar();
        if(m=='y'||m=='Y'){
            while((p1->next!=NULL)&&(id!=p1->id)){
                p2=p1;p1=p1->next;
            }
            if(id==p1->id){
                if(p1==top)
                    top=p1->next;
                else
                    p2->next=p1->next;
                free(p1);
            }
            printf("成功删除以上信息!\n");
        }
    }
}

```

```

        printf("请按任意键返回主菜单!\n");getch();
        break;
    case 3: printf("请按任意键返回!\n");getch();break;
    default :printf("您输入的的错误!\n 请重新选择\n");break;
    }
} while(choose!=1&&choose!=2&&choose!=3);
}
else{
    printf("没有信息，请先存入信息!\n");
    printf("请按任意键返回主菜单!\n");
    getch();
}
return top;
}

MARK *deletestuMarkall(MARK *top){
    int count=0;
    while(top!=NULL){
        count++;
        top=top->next;
    }
    printf("总共删除%d 条信息",count);
    return top;
}

void printscholarship(MARK *top){
    int i=0;
    printf("一等奖学金名单:\n");
    printf("姓名\t 学号\t 总分\n");
    while(top!=NULL){
        i++;
        printf("%s\t%d\t%.4lf\n",top->name,top->id,top->total);
        top=top->next;
        if(i>3) break;
    }
    if(i==3){
        printf("二等奖学金名单:\n");
        printf("姓名\t 学号\t 总分\n");
        while(top!=NULL){
            i++;
            printf("%s\t%d\t%.4lf\n",top->name,top->id,top->total);
            top=top->next;
            if(i>9)break;
        }
    }
    if(i==9){
        printf("三等奖学金名单:\n");
        printf("姓名\t 学号\t 总分\n");
        while(top!=NULL){
            i++;
            printf("%s\t%d\t%.4lf\n",top->name,top->id,top->total);
            top=top->next;
            if(i>15)break;
        }
    }
    printf("请按任意键返回!\n");
    getch();
}

```

```

MESSAGE *readfilestu(MESSAGE *head){
    char *filename;
    MESSAGE *add;
    FILE *stu;
    filename=(char *)malloc(sizeof(char));
    fflush(stdin);
    printf("请输入你要读取文件的名字!\n");
    gets(filename);
    filename=strcat(filename, ".txt");
    stu=fopen(filename, "r");
    if(stu==NULL){
        printf("文件中没有任何信息，请先存入信息!\n");
    }
    else{
        while(!feof(stu)){
            add=MES;
            if(add==NULL){
                printf("申请内存失败!\n");
                exit(1);
            }
            printf("姓名\t学号\t性别\t学院\t班级\n");
            fscanf(stu, "%s %d %s %s %d",
                add->name, &add->id, add->sex, add->collage, &add->Class);
            printf("%s\t%d\t%s\t%s\t%d\n", add->name, add->id, add->sex, add->collage, add->Class);
            add->next=head;
            head=add;
            add=add->next;
        }
        fclose(stu);
        printf("读取信息成功!\n");
    }
    printf("请按任何键返回主菜单!\n");
    getch();
    return head;
}

MARK *readfilestuMark(MARK *top){
    int i,k=0;
    char *filename;
    FILE *stu;
    MARK *add,*p1=top,*p2=top;
    filename=(char *)malloc(sizeof(char));
    fflush(stdin);
    printf("请输入你要读取文件的名字!\n");
    gets(filename);
    filename=strcat(filename, ".txt");
    stu=fopen(filename, "r");
    if(stu==NULL){
        printf("文件中没有任何信息，请先存入信息!\n");
    }
    else{
        while(!feof(stu)){
            add=MAR;
            if(add==NULL){
                printf("申请内存失败!\n");
                exit(1);
            }
            printf("姓名\t学号\n");

```

```

        fscanf(stu,"%s %d",add->name,&add->id);
        printf("%s\t%d\n",add->name,add->id);
        for(i=0;i<length;i++){
            printf("%s\t",subname[i]);
        }
        printf("\n");
        for(i=0;i<length;i++){
            fscanf(stu,"%f",&add->sub[i].score);
            printf("%.3f\t",add->sub[i].score);
            if(add->sub[i].score!=0){
                add->sub[k].subid=i+1;
                k++;
            }
        }
        add->sub[k].subid=-1;
        printf("总分\n");
        fscanf(stu,"%lf\n",&add->total);
        printf("%.4lf\n",add->total);
        add->next=NULL;
        if(!top){
            top=add;
        }
        else {
            //找到插入的位置
            while(p1){
                if(p1->total>=add->total){
                    p2=p1;
                    p1=p1->next;
                }
                else
                    break;
            }
            add->next=p1;
            p2->next=add;
        }
        add=add->next;
    }
    fclose(stu);
    printf("读取信息成功!\n");
}

printf("请按任何键返回主菜单!\n");
getch();
return top;
}

void savefilestu(MESSAGE *head){
    char *filename;
    FILE *stu;
    MESSAGE *p;
    if(head==NULL){
        printf("没有任何作息,请先存入信息!\n");
        printf("请按任意键返回主菜单!\n");
        getch();
    }
    else{
        p=head;
        filename=(char *)malloc(sizeof(char));

```

```

        fflush(stdin);
        printf("请输入你要读取文件的名字!\n");
        gets(filename);
        filename=strcat(filename, ".txt");
        stu=fopen(filename, "w");
        do{
            printf("姓名\t 学号\t 性别\t 学院\t 班级\n");//读入文件并且打印在屏幕上
            fprintf(stu, "%s\t%d\t%s\t%s\t%d\t", p->name, p->id, p->sex, p->collage, p->Class);
            printf("%s\t%d\t%s\t%s\t%d\n", p->name, p->id, p->sex, p->collage, p->Class);
            p=p->next;
        }while(p!=NULL);
        fclose(stu);
        printf("已经将信息保存到 STUDENT.txt 中!\n");
        printf("请按任何键返回主菜单!\n");
        getch();
    }
}

```

```

void savefilestuMark(MARK *top){
    int i,j,gple=0,k=0;
    char *filename;
    FILE *stu;
    MARK *p;
    p=MAR;
    if(p==NULL){
        printf("申请内存失败!\n");
        exit(1);
    }
    if(top==NULL){
        printf("没有任何作息,请先存入信息!\n");
        printf("请按任意键返回主菜单!\n");
        getch();
    }
    else{
        p=top;
        filename=(char *)malloc(sizeof(char));
        fflush(stdin);
        printf("请输入你要保存文件的名字!\n");
        gets(filename);
        filename=strcat(filename, ".txt");
        stu=fopen(filename, "a+");
        do{
            printf("姓名\t 学号\n");
            fprintf(stu, "%s\t%d\t", p->name, p->id);
            printf("%s\t%d\n", p->name, p->id);
            for(i=0; i<length; i++){
                printf("%s\t", subname[i]);
            }
            for(i=0; i<length; i++){
                j=p->sub[i].subid;
                if(j== -1){
                    while(k<length){
                        fprintf(stu, "0\t");
                        printf("0\t");
                        k++;
                    }
                    break;
                }
            }
        }
    }
}

```

```

        while(k<length){
            k++;
            if(j==k){
                fprintf(stu,"%0.3f\t",p->sub[k-1].score);
                printf("%0.3f\t",p->sub[k-1].score);
                break;
            }
            else{
                fprintf(stu,"0\t");
                printf("0 \t");
            }
        }
        printf("\n");
        printf("总分\n");
        fprintf(stu,"%0.4lf\n",p->total);
        printf("%0.4lf\n",p->total);
        p=p->next;
    }while(p!=NULL);
    fclose(stu);
    printf("已经将信息保存到%s 中!\n",filename);
}

printf("请按任何键返回主菜单!\n");
getch();
}

```

## 五、代码测试

1、程序运行主菜单如下:

学生管理系统

- 1 学生基本信息管理
- 2 学生成绩信息管理
- 3 文件操作
- 4 退出

2、此时选择相应选项会进入某个程序，例如选择2，回车，  
学生成绩信息管理

- 1 查找成绩信息
- 2 添加成绩信息
- 3 删除成绩信息
- 4 修改成绩信息
- 5 奖学金信息
- 6 返回

3、继续选择2，回车

请输入一个学生的学号(以输入0结束)

4、输入2007 回车

输入该学生的姓名

5、输入wang，回车

1 高数 2 高代 3 C语言 4 英语 5 工概 6 心健 7 思修 8 德育 9 体育 10 奖励

请选择你所选修的课程(输入-1结束选择)

6、输入1-1，回车

请输入该生高数的成绩:



- 7、输入 89 , 回车,  
请输入一个学生的学号(以输入 0 结束)
- 8、输入 0, 返回主界面, 如果选择 1, 则进入学生基本信息管理, 这里选择 2, 进入学生成绩信息管理系统的界面,  
1 个人信息  
2 全部信息  
请选择
- 9、选择 1, 回车  
请选择 (1 姓名 2 学号, 3 返回)
- 10、选择 1, 回车  
请输入要查找的名字
- 11、输入 wang, 回车  
姓名 学号  
Wang 2007  
高数  
89. 00
- 12、按任意键返回主菜单, 如果选择 1, 则进学生基本信息管理系统, 选择 4 , 则退出本系统, 回车  
Press any key to continue
- 13、按任意键则退出整个程序

## 六、思考题

- 1、为什么使用 malloc 申请内存, 用 free 释放内存? 如果不使用它们会用什么结果?
- 2、结构体中嵌套结构体的优缺点体现在哪些方面?
- 3、如何实现按不同科目对学生排序?

## 第二章 旅馆信息管理

本章要求设计一个旅店信息管理系统，以文件和数组完成对旅客和房间信息的管理，实现旅客的入住，换房和退房，查看旅客和房间信息，以及从文件读取相关信息，将相关信息储存到文件的功能。使用结构体来存放旅客和房间的信息；使用动态数组来实现信息的添加、删除等操作；用文件来保存信息。

### 一、实践目的

- 1、熟练结构体的定义和结构数组的使用。能对数组进行一系列的基本操作，如遍历，删除某一元素等。
- 2、熟悉文件的操作，对文件能够熟练进行读取，写入，追加等操作。
- 3、体会多文件编程的思想和工程的意义。

### 二、基本要求

- 1、本章要求实现旅客的入住，换房和退房，查看旅客和房间信息，以及从文件读取相关信息，将相关信息储存到文件的功能，从而实现旅馆信息管理的基本功能。
- 2、要求用两个结构体分别存储旅客和房间的信息。用一个枚举类型定义房间的类型。
- 3、使用结构数组来实现旅客入住，换房，退房三种行为。
- 4、能将文件中的信息读到结构体中，能将结构体的信息保存到文件中。
- 5、系统制作完成后应实现类似图 2.1 所示界面。

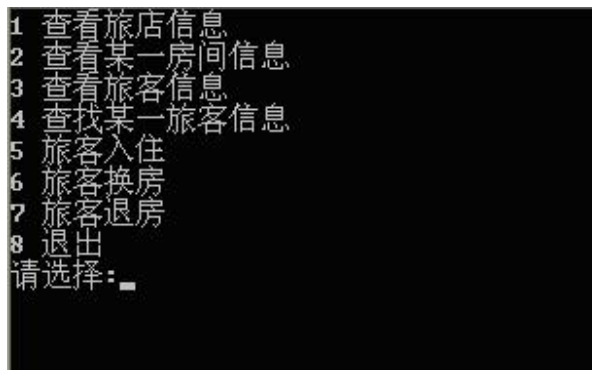


图 2.1 程序运行界面

### 三、算法分析

#### 1、数据结构

(1) 存放房间信息的结构体

```
typedef struct room{
    int RoomNumber;//房间号
    RoomType roomtype;//房间类型
    int CheckInNumber;//入住几人
    int RoomPrice;//房间价格
}Room;//房间相关的成员的定义
```

其中 RoomType 为自定义的枚举类型，其具体定义如下：

typedef enum RoomType{Single=1,Double=2,Triangle=3}RoomType;//房间类型  
分别表示单人房，双人房，三人房。

(2) 存放旅客信息的结构体

```
typedef struct guest{
    char Name[10];//姓名
    char sex[3];//性别
    char ID[20];//身份证号码
}Guest;//旅客的相关属性的定义
```

(3) 存放旅客与房间关系的信息结构体

```
typedef struct guestandroom{
    char GuestID[20];//旅客身份证号
    int GuestInRoom;//旅客入住的房间
    char ChickInTime[30];//住入的时间
    char ChickOutTime[30];//退房的时间
}GuestAndRoom;//房间相关的成员的定义
```

2、函数定义

文件分类和相关函数的简要说明如表 2.1 所示。

表 2.1 文件及函数说明

原文件	函数原型	功能说明
MainBody.c	void main()	总控函数
	void PrintMain()	打印菜单函数
	void HandleMain()	处理菜单函数
Hotel.c	void LoadHotelInfo(Room pararoom[FLOORNUMBER][])	从文件中读取相应的房间信息到房间结构体数组中，参数是一个 Room 结构体类型的二维数组。其中两个下标分别为楼层数和每个楼层的房间数
	void SaveHotelInfo(Room pararoom[FLOORNUMBER][])	将房间结构体数组中的相关信息保存到文件中，参数为一个 Room 结构体类型的二维数组
	int CheckFreeRoom(char Sex[])	检查是否有空房，Sex 是入住新人的性别，返回值为房间号
	char *JudgeRoomSex (int RoomNumber)	判定房间能让哪种性别的人入住，RoomNumber 是房间号，返回值是“男”，“女”或空值
	void ViewAllHotel()	打印所有房间入住情况
	void ViewOneRoom()	查看某一房间旅客信息
Guest.c	void LoadGuestInfo(int LocalGuestNumber)	从文件中读取现有旅客的信息到旅客结构体数组中，参数为要选择读取的旅客的数目
	void SaveGuestInfo(int LocalGuestNumber)	将现有旅客的信息从旅客结构体数组中保存到文件，参数为要选择保存的旅客的数目
	void ViewOneGuest()	打印某一旅客的信息
	void ViewAllGuest()	查看当前所有旅客的信息
	int FindOneGuest()	查找某一旅客，返回值是该旅客所在数组中的下标
	void AppendGuestInfo()	添加旅客信息
	int DelGuest(char name[],char ID[])	删除旅客信息，name 是删除顾客的姓名，ID 是删除顾客的身份证，返回值是根据住房类型获得的单日金

		额
	int CountGuest(Room pararoom[FLOORNUMBER][])	计算当前旅客的数量，pararoom 数组表示所有房间，返回值是现有旅客的数目
	int IsOldGuest(char NewID[])	判定是不是老顾客
	void AppendGuestHistoryInfo(int Guesti, int GARj)	将旅客的信息添加到历史记录中，Guesti 是要删除的旅客数组的索引，GARj 是 GAR 数组索引
GAR.c	void GuestCheckIn()	实现旅客的入住功能
	void GuestChangeRoom()	旅客换房
	void GuestCheckOut()	实现旅客的退房
	void AppendGARInfo()	添加旅客与房间信息
	void LoadGARInfo();	读取旅客与房间信息
	void SaveGARInfo()	保存旅客与房间信息
	int CheckRAGIndex(char ID[])	检验旅客对应的旅客与房间数组索引，ID 是旅客的身份证号码，返回值是 RAG 数组的索引
Structs.h		相关结构体和宏定义
		相关函数的声明
		包含的头文件

### 3、处理过程

#### (1) 实现入住过程

1) 输入住店人姓名，身份证，性别信息

```
printf("请输入旅客姓名:\n");
scanf("%s",ptoguest[GuestNumber].Name);
do{
    printf("请选择旅客性别:\n");
    printf("1 男\n2 女\n");
    scanf("%d",&NewGuestSexN);
    switch(NewGuestSexN){
        case 1:
            strcpy(ptoguest[GuestNumber].sex,"男");
            break;
        case 2:
            strcpy(ptoguest[GuestNumber].sex,"女");
            break;
    }
}
while(NewGuestSexN!=1&&NewGuestSexN!=2);
printf("请输入旅客身份证号码:\n");
scanf("%s",ptoguest[GuestNumber].ID);
```

2) 检查所有空房并打印

```
CheckSex=CheckFreeRoom(ptoguest[GuestNumber].sex);
```

3) 输入房间号码:

```
printf("请输入旅客选择入住房间号码:\n");
scanf("%d",&NewRoomNumber);
```

4) 更改房间状态

```
for(i=0;i<4;i++){
    for(j=0;j<5;j++){
        if(rooms[i][j].RoomNumber==NewRoomNumber){
```

```

rooms[i][j].CheckInNumber=rooms[i][j].CheckInNumber+1;
    }
}
}
5) 使 GuestNumber 递增
GuestNumber+=1;
6) 将相关信息 (入住时间, 旅客身份证, 旅客所住房间) 写入旅客与房间数组中
strcpy(GAR[GuestNumber-1].CheckInTime,ctime(&t));
printf("入住时间为%s\n",GAR[GuestNumber-1].CheckInTime);
strcpy(GAR[GuestNumber-1].GuestID,ptoguest[GuestNumber-1].ID);
GAR[GuestNumber-1].GuestInRoom=NewRoomNumber;
7) 将相关信息添加到文件中
AppendGuestInfo();
AppendGARInfo();
SaveHotelInfo(rooms);

```

#### (2) 实现退房过程

```

1) 输入退房旅客身份证:
printf("请输入旅客身份证:");
scanf("%s",ID);
2) 计算打折率:
discount=(IsOldGuest(ID)<=2)?1:(IsOldGuest(ID)<=4)?0.9:(IsOldGuest(ID)<=6)?0.8:0.7;
printf("此位顾客此次为第%d 次住店,打折率为:%f\n",IsOldGuest(ID)+1,discount);
3) 调用 DelGuest 函数, 返回每日房间费用:
money=(float)DelGuest(ID);
4) 根据入住时间和退房时间输入天数:
printf("请输入入住天数:\n");
scanf("%d",&Days);
5) 打印应付钱数:
money=money*Days*discount;
printf("应该付%f 元.\n",money);
6) 保存相关文件:
SaveGARInfo();
SaveGuestInfo();
SaveHotelInfo(rooms);

```

## 四、参考代码

```

/*MainBody.c*/

#include"Structs.h"
extern int GuestNumber;
int GuestNumber=0;//现有的旅客数
void main(void){
    LoadHotelInfo(rooms);
    LoadGuestInfo();
    LoadGARInfo();
    handlemenu();
}
void printMenu(){
    printf(
        "1 查看旅店信息\n"
        "2 查看某一房间信息\n"

```

```

        "3 查看旅客信息\n"
        "4 查找某一旅客信息\n"
        "5 旅客入住\n"
        "6 旅客换房\n"
        "7 旅客退房\n"
        "8 退出\n"
    );
}

void handlemenu(){
    int i=0;
    do{
        printMenu();
        printf("请选择:");
        scanf("%d",&i);
        fflush(stdin);
        switch(i){
            case 1:
                ViewAllHotel();
                printf("\n\n");
                break;
            case 2:
                ViewOneRoom();
                printf("\n\n");
                break;
            case 3:
                ViewAllGuest();
                printf("\n\n");
                break;
            case 4:
                ViewOneGuest();
                printf("\n\n");
                break;
            case 5:
                GuestCheckIn();
                printf("\n\n");
                break;
            case 6:
                GuestChangeRoom();
                printf("\n\n");
                break;
            case 7:
                GuestCheckOut();
                printf("\n\n");
                break;
            case 8:{
                SaveGARInfo();
                SaveGuestInfo();
                SaveHotelInfo(rooms);
                exit(0);
            }
        }
    }while(1);
}

/*Structs.c*/

#include<stdio.h>
#include<stdlib.h>

```

```

#include<conio.h>
#include<string.h>
#include<time.h>
#define FLOORNUMBER 4
#define ROOMNUMBER 5
typedef enum RoomType{Single=1,Double=2,Triangle=3}RoomType;//房间类型
typedef struct room{
    int RoomNumber;//房间号
    RoomType roomtype;//房间类型
    int CheckInNumber;//入住几人
    int RoomPrice;//房间价格
}Room;//房间相关的成员的定义
typedef struct guestandroom{
    char GuestID[20];//旅客身份证号
    int GuestInRoom;//旅客入住的房间
    char CheckInTime[30];//住入的时间
    char CheckOutTime[30];//退房的时间
}GuestAndRoom;//房间相关的成员的定义
typedef struct guest{
    char Name[10];//姓名
    char sex[3];//性别
    char ID[20];//身份证号码
}Guest;//旅客的相关属性的定义
void menu();
void LoadHotelInfo(Room pararoom[FLOORNUMBER][]);
void ViewAllHotel();
void SaveHotelInfo(Room pararoom[FLOORNUMBER][]);
void LoadGuestInfo();
void SaveGuestInfo();
void ViewOneGuest();
void ViewAllGuest();
void AppendGuestInfo();
void GuestCheckIn();
void GuestChangeRoom();
void GuestCheckOut();
void ChangeBookDay();
void AppendGARInfo();
void LoadGARInfo();
void SaveGARInfo();
void AppendGuestHistoryInfo(int Guesti,int GARj);
void handlemenu();
void printMenu();
void ViewOneRoom();
int IsOldGuest();
int DelGuest(char ID[]);
int CountGuest(Room pararoom[FLOORNUMBER][]);
int CheckRAGIndex(char ID[]);
int CheckFreeRoom(char Sex[]);
int FindOneGuest();
char *CheckRoomSex(int RoomNumber);
Room rooms[FLOORNUMBER][ROOMNUMBER];//存房间信息的结构数组
Guest ptoquest[55];//存旅客信息的数组
GuestAndRoom GAR[55];//存旅客入住房间的数组
/*GAR.c*/

#include"Structs.h"
extern int GuestNumber;
//向 gusetandroom 文件中添加信息

```

[illegible]



```

    }
}
return garindex;
}
//旅客入住
void GuestCheckIn(){
    int NewRoomNumber;
    int NewGuestSexN;
    int i;
    int j;
    int CheckSex=0;
    time_t t;
    t=time(NULL);
    printf("请输入旅客姓名:\n");
    scanf("%s",ptoguest[GuestNumber].Name);
    do{
        printf("请选择旅客性别:\n");
        printf("1 男\n2 女\n");
        scanf("%d",&NewGuestSexN);
        switch(NewGuestSexN){
            case 1:
                strcpy(ptoguest[GuestNumber].sex,"男");
                break;
            case 2:
                strcpy(ptoguest[GuestNumber].sex,"女");
                break;
        }
    }
    while(NewGuestSexN!=1&&NewGuestSexN!=2);
    printf("请输入旅客身份证号码:\n");
    scanf("%s",ptoguest[GuestNumber].ID);
    CheckSex=CheckFreeRoom(ptoguest[GuestNumber].sex);
    if(!CheckSex){
        //处理数组
        strcpy(ptoguest[GuestNumber].ID,"");
        strcpy(ptoguest[GuestNumber].Name,"");
        strcpy(ptoguest[GuestNumber].sex,"");
        return;
    }
    printf("请输入旅客选择入住房间号码:\n");
    scanf("%d",&NewRoomNumber);//如何保证选得是正确的呢
    //更改房间状态,新房间
    for(i=0;i<4;i++){
        for(j=0;j<5;j++){
            if(rooms[i][j].RoomNumber==NewRoomNumber){
                rooms[i][j].CheckInNumber=rooms[i][j].CheckInNumber+1;
            }
        }
    }
    GuestNumber+=1;//人数增加
    strcpy(GAR[GuestNumber-1].CheckInTime,ctime(&t));
    printf("入住时间为%s\n",GAR[GuestNumber-1].CheckInTime);
    strcpy(GAR[GuestNumber-1].GuestID,ptoguest[GuestNumber-1].ID);
    GAR[GuestNumber-1].GuestInRoom=NewRoomNumber;
    //将信息添加到文件中
    AppendGuestInfo();
    AppendGARInfo();
}

```

```

        SaveHotelInfo(rooms);
    }
    //旅客退房
    void GuestCheckOut(){
        float discount=0,money=0;
        int Days=0,i;
        char ID[20]="";
        do{
            printf("请输入旅客身份证:");
            scanf("%s",ID);
            for(i=0;i<=GuestNumber-1;i++){
                if(!strcmp(ID,ptoguest[i].ID)){
                    break;
                }
            }
            if(i==GuestNumber){
                printf("您输入的用户不存在！");
                i=-1;
            }
        }while(i!=-1);
        discount=(IsOldGuest(ID)<=2)?1:(IsOldGuest(ID)<=4)?0.9:(IsOldGuest(ID)<=6)?0.8:0.7;
        printf("此位顾客此次为第%d 次住店,打折率为:%f\n",IsOldGuest(ID)+1,discount);
        money=(float)DelGuest(ID);
        printf("请输入入住天数:\n");
        scanf("%d",&Days);
        money=money*Days*discount;
        printf("应该付%f 元.\n",money);
        SaveGARInfo();
        SaveGuestInfo();
        SaveHotelInfo(rooms);
    }
    //旅客换房
    //旅客换房，只能换到空房（结账）
    void GuestChangeRoom(){
        char name[10]="";
        char ID[19]="";
        int RAGindex=-1;
        int GuestIndex=-1;
        char choice='\0';
        int day=0;
        int i=0,j=0;
        int sum=0;
        int roomnumber=0;
        //输入旅客信息
        printf("请输入旅客的姓名:\n");
        gets(name);
        //找到旅客在旅客结构体中的下标
        printf("请输入需要换房的旅客的身份证: \n");
        gets(ID);
        RAGindex=CheckRAGIndex(ID);
        for(i=0;i<=GuestNumber;i++){
            if(!strcmp(ID,ptoguest[i].ID)){ //要是同名的就麻烦了
                printf("该旅客的信息如下\n"
                    "姓名\t性别\t入住的房间\t入住的时间\n");
                printf("%s\t%s\t%d\t%s\n",ptoguest[i].Name,ptoguest[i].sex,GAR[RAGindex].GuestIn
                    Room,GAR[RAGindex].CheckInTime);
                GuestIndex=i;
                break;
            }
        }
    }

```

```

    }
}
printf("确认以上信息? (y/n)\n");
choice=getchar();
if(choice=='N'||choice=='n'){
    return;
}
//结账
printf("请输入该旅客的入住天数: \n");
scanf("%d",&day);
for(i=0;i<FLOORNUMBER;i++){
    for(j=0;j<ROOMNUMBER;j++){
        if(rooms[i][j].RoomNumber==GAR[RAGindex].GuestInRoom){
            sum=day*rooms[i][j].RoomPrice;
            break;
        }
    }
    if(sum){
        break;
    }
}
printf("请旅客交纳前面的房款%d.\n",sum);
//打印能换入的房间
CheckRAGIndex(ptoguest[GuestIndex].sex);
printf("请输入要换入的房间号: \n");
scanf("%d",&roomnumber);
//放入
if(roomnumber==GAR[RAGindex].GuestInRoom){
    printf("不能换入原房间");
    return;
}
//状态更改
//原房间
GAR[RAGindex].GuestInRoom=roomnumber;
rooms[i][j].CheckInNumber--;
for(i=0;i<FLOORNUMBER;i++){
    for(j=0;j<ROOMNUMBER;j++){
        if(rooms[i][j].RoomNumber==roomnumber){
            rooms[i][j].CheckInNumber++;
            break;
        }
    }
}
SaveGARInfo();
SaveGuestInfo();
SaveHotelInfo(rooms);
}
Guests.c:
#include"Structs.h"
extern int GuestNumber;
//添加旅客信息
void AppendGuestInfo(){
    FILE *fp;
    if((fp=fopen("guest.txt","a"))==NULL){
        printf("文件读取失败");
        exit(0);
    }
    fprintf(fp,"%s\t%s\t%s\t\n",ptoguest[GuestNumber-1].Name,ptoguest[GuestNumber-1].sex,ptoguest[GuestNumber-1].ID);
}

```

```

        fclose(fp);
    }

int CountGuest(Room pararoom[FLOORNUMBER][ROOMNUMBER]){
    int iFloor=0;
    int jRoom=0;
    int sum=0;
    for(iFloor=0;iFloor<FLOORNUMBER;iFloor++){
        for(jRoom=0;jRoom<ROOMNUMBER;jRoom++){
            if(pararoom[iFloor][jRoom].CheckInNumber>0){
                sum+=pararoom[iFloor][jRoom].CheckInNumber;
            }
        }
    }
    return sum;
}

//载入客人信息
void LoadGuestInfo(){
    FILE *fp;
    int iCount=0;
    char temp[10]="";
    if((fp=fopen("guest.txt","r"))==NULL){
        printf("文件读取失败!\n");
        exit(0);
    }
    GuestNumber=CountGuest(rooms);
    fscanf(fp,"%s %s %s",temp,temp,temp);
    for(iCount=0;iCount<GuestNumber;iCount++){
        fscanf(fp,"%s\t%s\t%s",ptoguest[iCount].Name,ptoguest[iCount].sex,ptoguest[iCount].ID);
        //strcpy(GAR[iCount].GuestID,ptoguest[iCount].ID);
    }
    fclose(fp);
}

//储存旅客信息
void SaveGuestInfo(){
    FILE *fp;
    int iCount=0;
    if((fp=fopen("guest.txt","w"))==NULL){
        printf("文件读取失败");
        exit(0);
    }
    fprintf(fp,"%s\t%s\t%s\n","姓名","性别","身份证");
    for(iCount=0;iCount<GuestNumber;iCount++){
        fprintf(fp,"%s\t%s\t%s\n",ptoguest[iCount].Name,ptoguest[iCount].sex,ptoguest[iCount].ID);
    }
    fclose(fp);
}

//查找旅客信息
void ViewOneGuest(){
    int i=FindOneGuest();
    int GARNUM;
    if(i!=-1){
        GARNUM=CheckRAGIndex(ptoguest[i].ID);
        printf("姓名\t性别\t身份证\t房间\t入住时间\n");
        printf("%s\t%s\t%s\t%d\t%s\n",ptoguest[i].Name,ptoguest[i].sex,ptoguest[i].ID,GAR[GARNUM].GuestInRoom,GAR[GARNUM].CheckInTime);
        return;
    }
    else{
        printf("没有此用户信息! \n");
    }
}

```

```

        return;
    }
}
//查找旅客信息可以做成一个函数
int FindOneGuest(){
    int i;
    char GuestID[25];
    printf("\n 请输入旅客的身份证:\n");
    scanf("%s",GuestID);
    fflush(stdin);
    for(i=0;i<GuestNumber;i++){
        if(!strcmp(GuestID,ptoguest[i].ID)){
            return i;
        }
    }
    printf("没有此名顾客! ");
    return -1;
}
//查看所有旅客的信息
void ViewAllGuest(){
    int i=0;
    int GARNUM;
    printf("-----所有旅客信息-----\n");
    printf("姓名\t性别\t身份证\t房间\t入住时间\n");
    for(i=0;i<GuestNumber;i++){
        GARNUM=CheckRAGIndex(ptoguest[i].ID);//找下标
        printf("%s\t%s\t%s\t%d\t%s\t\n",ptoguest[i].Name,ptoguest[i].sex,ptoguest[i].ID,GAR[
NUM].GuestInRoom,GAR[GARNUM].CheckInTime);
    }
    printf("打印结束!\n");
}
//判断是否是以前的旅客
int IsOldGuest(char NewID[]){
    FILE *fp;
    char temp[20]="";
    char name[10]="";
    char ID[20]="";
    int Room=0;
    char CheckInTime[30]="";
    char CheckOutTime[30]="";
    int a=0;
    if((fp=fopen("guesthistory.txt","r"))==NULL){
        printf("文件读取失败");
        exit(0);
    }
    fscanf(fp,"%s %s %s %s %s",temp,temp,temp,temp,temp);
    while(!feof(fp)){
        fscanf(fp,"%s %s %d %s %s",name,ID,&Room,CheckInTime,CheckOutTime);
        if(!strcmp(NewID,ID)){
            a++;
        }
    }
    return a;
}
//添加新的历史记录
void AppendGuestHistoryInfo(int Guesti,int GARj){
    FILE *fp;
    if((fp=fopen("guesthistory.txt","a"))==NULL){

```

```

        printf("文件读取失败");
        exit(0);
    }
    fprintf(fp,"%s\\t%s\\t%d\\t\\t%s\\t\\t%s\\n",ptoguest[Guesti].Name,ptoguest[Guesti].ID,GAR[Guesti].GuestInRoom,GAR[Guesti].CheckInTime,GAR[Guesti].CheckOutTime);
}
//打印入住记录
void ViewOldGuest(){
    FILE *fp;
    char name[10];
    int ID;
    int Room;
    char CheckInTime[30];
    char CheckOutTime[30];
    if((fp=fopen("guesthistory.txt","r"))==NULL){
        printf("文件读取失败");
        exit(0);
    }
    printf("姓名\\t  身份证\\t  房间号\\t  入住时间\\t  退房时间\\t\\n");
    do{
        fscanf(fp,"%s %d %d %s %s\\n",name,&ID,&Room,CheckInTime,CheckOutTime);
        printf("%s\\t %d\\t %d\\t %s\\t %s\\t\\n",name,ID,Room,CheckInTime,CheckOutTime);
    }
    while(!feof(fp));
}
//删除 Guest 数组中的一个信息,同时删除对应的 GAR
int DelGuest(char ID[]){
    int i=0,j=0,GARj=0,roomi=0,roomj=0;
    int money=0;
    int out=1;
    time_t t;
    t=time(NULL);
    for(i=0;i<GuestNumber;i++){
        if(!strcmp(ID,ptoguest[i].ID)){
            GARj=CheckRAGIndex(ptoguest[i].ID);
            printf("入店时间为%s\\n",GAR[GARj].CheckInTime);
            strcpy(GAR[GARj].CheckOutTime,ctime(&t));
            printf("退店时间为%s\\n",GAR[GARj].CheckOutTime);
            for(roomi=0;roomi<4&&out;roomi++){
                for(roomj=0;roomj<5;roomj++){
                    if(GAR[GARj].GuestInRoom==rooms[roomi][roomj].RoomNumber){
                        money=rooms[roomi][roomj].RoomPrice;
                        rooms[roomi][roomj].CheckInNumber=rooms[roomi][roomj].CheckInNumber-1;
                        SaveHotelInfo(rooms);
                        out=0;
                        break;
                    }
                }
            }
            AppendGuestHistoryInfo(i,GARj);
            GuestNumber=GuestNumber-1;
            for(j=i;j<GuestNumber;j++){
                ptoguest[j]=ptoguest[j+1];
            }
            for(j=GARj;j<GuestNumber;j++){
                GAR[j]=GAR[j+1];
            }
            break;
        }
    }
}

```

```

    }
}
return money;
}
Hotels.c:
#include "Structs.h"
extern int GuestNumber;
//检查房间应住的性别
char *CheckRoomSex(int RoomNumber){
    int i,j;
    for(i=0;i<GuestNumber;i++)//添加那有问题 GuestNumber{
        if(RoomNumber==GAR[i].GuestInRoom){//得到房间客人所在数组的下标
            for(j=0;j<GuestNumber;j++){
                if(!strcmp(GAR[i].GuestID,ptoguest[j].ID))//找到对应的客人
                    return ptoguest[j].sex;
            }
            break;
        }
    }
    return NULL;
}
//检查空房
int CheckFreeRoom(char Sex[]){
    int i,j,find=0;
    char roomsex[3];
    printf("能入住的房间:\n");
    for(i=0;i<4;i++){
        for(j=0;j<5;j++){
            if(!rooms[i][j].CheckInNumber){//房间为空
                printf("%d\t",rooms[i][j].RoomNumber);
                find=1;
            }
            else{//房间有人
                strcpy(roomsex,CheckRoomSex(rooms[i][j].RoomNumber));
                if((
                    rooms[i][j].roomtype>rooms[i][j].CheckInNumber
                    //入住的人数少于房间类型
                    && (strcmp(Sex,roomsex)==0)//入住的是相同性别人
                ))
                {
                    printf("%d\t",rooms[i][j].RoomNumber);
                    find=1;
                }
                else{
                    printf("\t");
                }
            }
        }
        printf("\n");
    }
    if(!find){
        printf("没有合适房间!\n 房间已满.\n");
    }
    return find;
}
//载入所有房间入住情况信息
void LoadHotelInfo(Room innerroom[FLOORNUMBER][ROOMNUMBER]){
    FILE *fp;

```

```

int iFloor=0;
int jRoom=0;
char temp[9];
if((fp=fopen("hotel.txt","r"))==NULL){
    printf("文件读取失败");
    exit(0);
}
fscanf(fp,"%s %s %s %s\n",temp,temp,temp,temp);
for(iFloor=0;iFloor<FLOORNUMBER;iFloor++){
    for(jRoom=0;jRoom<ROOMNUMBER;jRoom++){
        fscanf(fp,"%d %d %d
%d\n",&innerroom[iFloor][jRoom].RoomNumber,&innerroom[iFloor][jRoom].roomtype,&innerroom[iFloor][jRoom].CheckInNumber,&innerroom[iFloor][jRoom].RoomPrice);
    }
    fclose(fp);
}
//打印所有房间入住情况
void ViewAllHotel(){
    int i,j;
    printf("房间号码\t 房间类型\t 入住人数\t 房间价格\t\n");
    for(i=0;i<4;i++){
        for(j=0;j<5;j++){
            printf(" %d\t",rooms[i][j].RoomNumber);
            switch(rooms[i][j].roomtype){
                case 1:
                    printf(" 单人房\t ");break;
                case 2:
                    printf(" 双人房\t ");break;
                case 3:
                    printf(" 三人房\t ");break;
            }
            printf("%d\t%d\t%d\t\n",rooms[i][j].CheckInNumber,rooms[i][j].RoomPrice);
        }
        printf("\n");
    }
}
//储存所有房间信息
void SaveHotelInfo(Room pararoom[FLOORNUMBER][ROOMNUMBER]){
    FILE *fp;
    int iFloor=0;
    int jRoom=0;
    if((fp=fopen("hotel.txt","w"))==NULL){
        printf("文件读取失败");
        exit(0);
    }
    fprintf(fp,"房间号\t 房间类型\t 入住人数\t 房间价格\t\n");
    for(iFloor=0;iFloor<FLOORNUMBER;iFloor++){
        for(jRoom=0;jRoom<ROOMNUMBER;jRoom++){
            fprintf(fp,"
%d\t%d\t%d\t%d\t\n",pararoom[iFloor][jRoom].RoomNumber,pararoom[iFloor][jRoom].roomtype,pararoom[iFloor][jRoom].CheckInNumber,pararoom[iFloor][jRoom].RoomPrice);
        }
        fprintf(fp,"\n");
    }
    fclose(fp);
}
//打印某一房间情况
void ViewOneRoom(){

```



```

int RoomNum,i,j,IsNull=0;
printf("请输入房间号码: ");
scanf("%d",&RoomNum);
for(i=0;i<GuestNumber;i++){
    if(RoomNum==GAR[i].GuestInRoom){
        for(j=0;j<GuestNumber;j++){
            if(strcmp(ptoguest[j].ID,GAR[i].GuestID)==0){
                printf(" 姓名 :%s\t 身份证 :%s\t 性别 :%s\t 入住时
间:%s\n",ptoguest[j].Name,ptoguest[j].ID,ptoguest[j].sex,GAR[i].CheckInTime);
                IsNull=1;
            }
        }
    }
}
if(!IsNull){
printf("房间为空! \n");
}
}

```

## 五、代码测试

1、当程序运行时，主菜单显示如下：

```

*****
1 查看旅店信息
2 查看某一房间信息
3 查看所有旅客信息
4 查找某一旅客信息
5 旅客入住
6 旅客换房
7 旅客退房
8 退出
*****

```

请选择:

2、此时选择相应选项会进入下一级菜单，例如选择5 旅客入住，回车：

```

请输入旅客姓名：
小刚
请选择性别：
1 男
2 女
选择 1
请输入旅客身份证号码：
20746354637823
能入住的房间：
101 102 103 104 105
    202 203 204 205
        302 303 304 305
            401 402 403 404 405
                501 502 503 504 505

```

请选择入住号码:

101

入住时间为 Sat May 10 16:32:20 2008

*返回主菜单*

### 3、查看某一房间信息

选择2

请输入房间号码:

101

姓名: 小刚    身份证: 20746354637823    性别: 男    入住时间: Sat May 10 16:32:20 2008

*返回主菜单*

### 4、旅客换房

选择6

请输入旅客姓名:

闫振南

请输入需要换房的旅客的身份证:

555

该旅客的信息如下:

姓名	性别	入住房间	入住时间
闫振南	男	404	Fri May 16 17:58:22 2008

请输入该旅客的入住天数:

4

请缴纳前面的房款 160

请输入换入房间的号码:

101

*返回主菜单*

### 5、旅客退房

选择7

请输入退房旅客的身份证:

20746354637823

此位顾客第 1 次住店, 打折率为 1.000000

此位客人入店时间为 Sat May 10 16:32:20 2008

退店时间为 Sat May 10 16:34:29 2008

请输入入住天数

1

应该付 100.000000 元

*返回主菜单*

检查各个文件, 查看各个文件信息是否正确被删除, 历史文件是否正确被添加。  
其他功能也以此方法进行测试。

## 六、思考题

- 1、如何根据入住日期和退房日期直接计算旅客入住天数?
- 2、上述功能选择一次只能对一个旅客进行处理, 如何同时对多个旅客进行处理?
- 3、本章中用数组来处理旅客的信息, 如何用链表来实现?

## 第三章 餐厅信息管理

本章要求设计一个餐厅管理程序，要求实现客户点菜、客户结账和餐厅设备维护功能，每一种功能下面还需要分为许多小模块；使用各自不同的结构体来储存各种信息；使用链表来实现订单信息的创建、客户结账、账目管理等操作；使用文件来保存数据，下次运行时可以从文件中自动读取数据。

### 一、实践目的

- 1、掌握带头节点的链表的工作原理和处理方式。
- 2、会使用 malloc、free 等函数对链表进行创建、增加、删除等操作，会遍历链表以实现查询等操作。
- 3、加深理解模块化的编程思想，将一个程序划分成不同的函数来编写，掌握函数之间有效的调用关系。
- 4、会使用 C 语言对文件进行读取、修改等操作，掌握二进制文件和文本文件之间的区别。
- 5、了解餐厅管理过程中所需要处理的信息以及相关的处理方法。

### 二、基本要求

- 1、本程序要求实现客户点菜的过程、客户结账、账目的管理、餐厅系统的维护四大功能模块，每个功能模块又分别对应一些不同操作子模块，从而完成一个餐厅信息管理系统功能。
- 2、要求使用三种不同的结构体来分别存储餐桌、菜以及订单信息。
- 3、使用链表来实现创建客户订单与客户结账等操作。
- 4、使用文本文件完成数据的存储与读取，完成账单的管理。
- 5、系统制作完成后应实现类似图 3.1 所示界面。



图 3.1 程序运行界面

### 三、算法分析

#### 1、数据结构

##### (1) 订单结构体

```
typedef struct Order{  
    int Table_ID;//记录餐桌号  
    int Dish_ID[N];//记录菜的编号
```

```
    struct Order *next;
}ORDER;
(2) 餐桌结构体
typedef struct Table{
    int  Table_ID;//餐桌编号
    int  Capacity;//餐桌最多能容纳的人数
    int  Table_State;//1 有人在用,0 没人用
}TABLE;
(3) 菜结构体
typedef struct Dish{
    int  Dish_ID;//菜的编号
    char Dish_Name[15];//菜名
    float Dish_Price;//价格
}DISH;
TABLE tb[H];//声明餐桌结构体数组
DISH dish[D];//声明菜的结构体数组
函数定义与说明如表 3.1 所示。
```

表 3.1 文件及函数说明

源文件	函数原型	功能说明
restaurant.c	void Table_Init(void)	餐桌信息初始化
	void Load_Dish(void)	从文件中读取菜谱
	void Dish_Menu(void)	打印菜单
	void Display(int)	打印菜信息，参数为菜的编号
	int search(void)	查找匹配型号的餐桌，返回餐桌号
	ORDER*CreateOne(int )	创建一个客户节点，参数为餐桌编号，返回订单节点指针
	ORDER*Dish_Order(ORDER*, ORDER*)	点菜，参数分别为订单头指针和客户节点指针，返回订单头指针
	void Save_Inform (ORDER*, int)	消费历史记录，参数分别为客户节点指针和点菜的总数
	void saveInList(float )	将客户消费金额存入账目，参数为消费额
	ORDER*Payment (ORDRER*)	客户结账，参数为订单头指针，返回订单头指针
	int Pass_Word(void)	密码验证，返回密码验证的结果 1 或 0
	void ModifyPW()	修改账目密码
	void List_Management(void)	账本管理
	void Observe(void)	查看账目
	void ListMenu(void)	账目管理菜单
	void Main_Menu(void)	显示系统菜单
	void Menu_select(void)	完成系统各项功能
	void Get_Rank(void)	对菜进行统计排名

### 3、处理过程

#### (1) 点菜功能的实现

先调用 search()函数，查找客户所需要的餐桌，找到后调用 Dish\_Menu()函数打印菜谱，接着调用 CreateOne(float)函数创建一个订单节点，最后调用 Dish\_Order()函数，将新创建的订单链到链表表尾。至此，实现了客户点菜的功能。

#### (2) 结账功能的实现

先让客户输入自己就餐的餐桌编号，根据编号查找客户消费的具体情况，并调用 Display()函数打印客户点的菜的信息，接着调用 Save\_Inform()函数将订单信息写入历史记录文本文件中进行保存，作为统计的基础，然后调用和 saveInList()函数将客户消费总额和消费的具体时间存入名为“账本”的文件中，作为账目管理的基础。至此，实现了客户结账的功能模块。

#### (3) 账目管理

调用 void List\_Management()函数并且通过密码验证，进入 ListMenu()函数。选择功能 1，调用 ModifyPW()函数进行密码修改；选择功能 2，调用 Observe()函数查看账单；选择 3，则返回系统主菜单。

#### (4) 统计菜排名

通过调用 Get\_Rank()函数，打印近期比较受欢迎的四道菜。

```
while(!feof(fp)){
    fscanf(fp,"%d\n",&n);
    i=0;
    while(i!=n){
        fscanf(fp,"%d%s%s",&m,s,s);
        for(j=0;j<10;j++){
            if(m==dish[j].Dish_ID){
                count[j]++;
                break;
            }
        }
        i++;
        if(i==n)
            fscanf(fp,"%s%s%s%s%s%s",s,s,s,s,s);
    }
}
printf("菜的受欢迎程度如下: \n");
for(i=0;i<10;i++){
    printf("\n%d 菜: ",i+1);
    for(j=0;j<count[i];j++)
        printf(" * ");
}
```

### 四、参考代码

```
#include <stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#include<time.h>
#define N 10
#define D 10
```

```

#define H 10
// #define PASSWORD 123
#define ASK (ORDER*)malloc(sizeof(ORDER))
#define MaxCapacity 4
// 定义订单结构体
typedef struct Order{
    int Table_ID;//记录餐桌号
    int Dish_ID[N];//记录菜的编号
    struct Order *next;
}ORDER;

// 定义餐桌结构体
typedef struct Table{
    int Table_ID;// 餐桌编号
    int Capacity;// 餐桌最多能容纳的人数
    int Table_State;//1 有人在用,0 没人用
}TABLE;

// 定义菜的结构体
typedef struct Dish{
    int Dish_ID;//菜的编号
    char Dish_Name[10];//菜名
    float Dish_Price;//价格
}DISH;
TABLE tb[H];//餐桌结构体数组
DISH dish[D];//菜的结构体数组
int PASSWORD=123;//密码设置
// 函数定义
void Load_Dish(void);//从文件中读取菜谱
ORDER*CreateOne(int);//创建一个订单节点
void Dish_Menu(void);//显示菜谱
void Table_Init(void);//餐桌信息初始化
int search(void);//查找匹配的餐桌
ORDER* Dish_Order(ORDER *, ORDER *);//将节点添加到主链
void Display(int);//显示菜的信息
void saveInList(float);//将客户消费额写入账单
void Save_Inform(ORDER *, int);//将已结账订单信息写入历史记录
ORDER *Payment(ORDER *);//结账
void ModifyPW(void);//密码修改
int Pass_Word(void);//密码验证
void Observe(void);// 账目查询
void ListMenu(void);//账单管理菜单
void List_Management(void);//账单管理
void Main_Menu(void);//主菜单
void Get_Rank(void);//统计
void Menu_select(void);//选择功能
// 主函数
main(){
    Table_Init();
    Menu_select();
}
// 系统主菜单
void Main_Menu(){
    printf("\n*****\n");
    printf("餐厅服务系统\n");
    printf("*****\n");
    printf("1.点菜\n");
    printf("2.客户结账\n");
}

```

```

        printf("3.账目管理\n");
        printf("4.餐厅统计\n");
        printf("5.退出系统\n");
    }
//选择功能
void Menu_select(void){
    ORDER* head;
    int choose;
    int result;
    head=NULL;
    system("cls");
    do{
        Load_Dish();
        Main_Menu();
        printf("请选择:\n");
        fflush(stdin);
        scanf("%d",&choose);
        //system("cls");
        switch(choose){
            case 1: result=search();
                    if(result==1)
                        printf("您可以到%d 号餐桌就餐\n",result);
                    else
                        printf("您和您的朋友可以到%d 号餐桌就餐\n",result);
                        Dish_Menu();
                        head=Dish_Order(head,CreateOne(result));
                        //system("cls");
                        break;
            case 2: head=Payment(head);
                    break;
            case 3: List_Management();
                    break;
            case 4: Get_Rank();
                    break;
            case 5: exit(1);
                    break;
        }
    }while(1);
}
//从文件中读取菜谱
void Load_Dish(void){
    FILE *fp;
    int i;
    fp=fopen("dish_name.txt","r");
    for(i=0;i<D;i++)
    {
        fscanf(fp,"%d\t%s\t%f\n",&dish[i].Dish_ID,dish[i].Dish_Name,&dish[i].Dish_Price);
    }
}
//餐桌初始化
void Table_Init(){
    tb[0].Capacity=1;tb[0].Table_ID=1;
    tb[1].Capacity=1;tb[1].Table_ID=2;
    tb[2].Capacity=2;tb[2].Table_ID=3;
    tb[3].Capacity=2;tb[3].Table_ID=4;
    tb[4].Capacity=2;tb[4].Table_ID=5;
    tb[5].Capacity=3;tb[5].Table_ID=6;
    tb[6].Capacity=4;tb[6].Table_ID=7;
}

```

```

        tb[7].Capacity=4;tb[7].Table_ID=8;
        tb[8].Capacity=4;tb[8].Table_ID=9;
        tb[9].Capacity=4;tb[9].Table_ID=10;
    }
    //显示菜谱
    void Dish_Menu(void){
        int i;
        printf("*****欢迎选购本店菜*****\n");
        printf("菜编号\t菜名\t菜单价\n");
        for(i=0;i<D;i++){
            printf("%d\t%s\t%f\n",dish[i].Dish_ID,dish[i].Dish_Name,dish[i].Dish_Price);
        }
    }
    //查找有没有匹配的餐桌
    int search(void){
        int Cust_Num;
        int i;
        printf("请输入来客数量:\n");
        scanf("%d",&Cust_Num);
        if(MaxCapacity<Cust_Num){
            printf("抱歉,本店没有容纳%d 的餐桌!",Cust_Num);
            return -1;
        }
        while(Cust_Num<=MaxCapacity){
            for(i=0;i<H;i++){
                if(tb[i].Table_State==0){
                    if(tb[i].Capacity==Cust_Num)
                        return (tb[i].Table_ID);
                }
            }
        }
        printf("抱歉,现在没有%d 人桌",Cust_Num);
        Cust_Num+=1;
    }
    printf("餐桌已满, 请客人稍等一会。");
    return -1;
}
//创建一份订单
ORDER*CreateOne(int result){
    ORDER*p;
    int i;
    p=ASK;
    if(result!=-1){
        p->Table_ID=result;
        for(i=0;i<D;i++){
            printf("请输入菜的编号,按 0 结束输入:");
            scanf("%d",&p->Dish_ID[i]);
            if(p->Dish_ID[i]<0||p->Dish_ID[i]>10){
                printf("超出菜谱范围,请重新输入\n");
                i--;
            }
            else if(p->Dish_ID[i]==0)
                break;
            else
                p->Dish_State[i]=0;
        }
        tb[result-1].Table_State=1;
    }
    p->next=NULL;
    return p;
}

```



```

}
//添加到主链
ORDER* Dish_Order(ORDER *head,ORDER *p){
    ORDER *p1;
    p1=head;
    if(p1!=NULL){
        if(p!=NULL){
            while(p1->next!=NULL){
                p1=p1->next;
            }
            p1->next=p;
            printf("订单创建成功\n");
        }
        else
            printf("订单创建失败\n");
    }
    else{
        if(head==NULL&&*&p!=NULL)
            head=p;
    }
    return head;
}
//根据菜的编号打印一道菜的信息
void Display(int ID){
    int i=0;
    for(i=0;i<D;i++){
        if(dish[i].Dish_ID==ID){
            printf("%d\t%s\t%f\n",dish[i].Dish_ID,dish[i].Dish_Name,dish[i].Dish_Price);
            break;
        }
    }
}
//将消费额写入账单
void saveInList(float pay){
    FILE*fp;
    time_t timer;
    timer=time(NULL);
    if((fp=fopen("账本.txt","a"))==NULL){
        printf("操作失败\n");
        exit(1);
    }
    fprintf(fp,"%f\t%s\n",pay,ctime(&timer));//有问题 系统时间
    printf("账本保存成功\n");
    fclose(fp);
}
//将已结账客户写入历史记录
void Save_Inform(ORDER *p,int m){
    FILE *fp;
    time_t timer;
    int i=0;
    timer=time(NULL);
    if((fp=fopen("历史记录.txt","a"))==NULL){
        printf("操作失败\n");
        exit(1);
    }
    fprintf(fp,"%d\n",m);
    while(p->Dish_ID[i]>0){
        fprintf(fp,"%d\t%s\t%f\n",dish[p->Dish_ID[i]-1].Dish_ID,dish[p->Dish_ID[i]-1].Dish_Name

```

```

,dish[p->Dish_ID[i]-1].Dish_Price);
    i++;
}
fprintf(fp,"%s",ctime(&timer));
printf("历史记录保存成功\n");
fclose(fp);
}
//结账
ORDER *Payment(ORDER *head){
    int i=0;
    int count=0;
    float pay=0.0;
    float Pay;
    int ID;
    ORDER *p,*p1;
    p=head;
    printf("请输入您的餐桌号\n");
    fflush(stdin);
    scanf("%d",&ID);
    while(p!=NULL){
        if(p->Table_ID==ID){
            printf("您点菜的情况如下:\n");
            printf("编号\t菜名\t价格\n");
            while(p->Dish_ID[i]!=0){
                Display(p->Dish_ID[i]);
                pay+=dish[p->Dish_ID[i]-1].Dish_Price;
                i++;
                count++;
            }
            printf("您一共点了%d道菜\n",count);
            printf("您本次的消费额为%f元\n",pay);
            printf("您实际的付款:\n");
            scanf("%f",&Pay);
            if(Pay>pay)
                printf("找您%f\n",Pay-pay);
            printf("谢谢您的惠顾,欢迎下次光临\n");
            if(count>0){
                saveInList(pay);
                Save_Inform(p,count);
            }
            tb[p->Table_ID].Table_State=0;
            break;
        }
        else{
            p1=p;
            p=p->next;
        }
    }
    if(p==head)
        head=head->next;
    else
        p1->next=p->next;
    free(p);
    return head;
}
//修改密码
void ModifyPW(){
    int password;
    printf("请输入新的密码\n");

```

```

        scanf("%d",&password);
        PASSWORD=password;
        printf("密码修改成功\n");
    }
//查看账目
void Observe(){
    FILE *fp;
    float pay;
    char str[25];
    int i=0;
    int j=0;
    if((fp=fopen("账本.txt","r"))==NULL){
        printf("操作失败\n");
        exit(1);
    }
    printf("消费金额\t 消费时间\n");
    while(!feof(fp)){
        i=0;
        j=0;
        fscanf(fp,"%f\t",&pay);
        while(i<=5&&j<25){
            fscanf(fp,"%c",&str[j]);
            j++;
            if(str[j]==' ')
                i++;
        }
        fscanf(fp,"\n\n");
        i=0;
        j=0;
        printf("%f\t",pay);
        while(i<=5&&j<25){
            printf("%c",str[j]);
            j++;
            if(str[j]==' ')
                i++;
        }
        printf("\n");
    }
    fclose(fp);
}
//账单管理菜单
void ListMenu(){
    int choice;
    printf("1.修改密码\n");
    printf("2.查看账本\n");
    printf("3.返回\n");
    do{
        printf("请选择:\n");
        scanf("%d",&choice);
        switch(choice) {
            case 1:ModifyPW();
                break;
            case 2:Observe();
                break;
            case 3:Menu_select();
                break;
            default:printf("没有该功能项\n");
        }
    }
}

```

```

        }while(1);
    }
//账单管理
void List_Management(){
    FILE *fp;
    if((fp=fopen("账本.txt","r"))==NULL){
        printf("操作失败\n");
        exit(1);
    }
    if(Pass_Word())
        ListMenu();
}
//获得近期各菜的受欢迎程度
void Get_Rank(void){
    FILE *fp;
    int n,i,m=0,j;
    int count[10]={0},t=0;
    char s[16]="",a[10];
    fp=fopen("历史记录.txt","r");
    while(!feof(fp)){
        fscanf(fp,"%d\n",&n);
        i=0;
        while(i!=n){
            fscanf(fp,"%d%s%s",&m,s,s);
            for(j=0;j<10;j++){
                if(m==dish[j].Dish_ID){
                    count[j]++;
                    break;
                }
            }
            i++;
            if(i==n)
                fscanf(fp,"%s%s%s%s%s%s",s,s,s,s,s);
        }
    }
    printf("菜的受欢迎程度如下: \n");
    for(i=0;i<10;i++){
        printf("\n%d 菜: ",i+1);
        for(j=0;j<count[i];j++)
            printf(" * ");
    }
    for(i=0;i<9;i++){
        for(j=0;j<9-i;j++){
            if(count[j]<count[j+1]){
                t=count[j];
                strcpy(a,dish[j].Dish_Name);
                count[j]=count[j+1];
                strcpy(dish[j].Dish_Name,dish[j+1].Dish_Name);
                count[j+1]=t;
                strcpy(dish[j+1].Dish_Name,a);
            }
        }
    }
    printf("\n 比较受顾客欢迎的四个菜是: \n");
    for(i=0;i<4;i++){
        printf("%s\n",dish[i].Dish_Name);
    }
}

```

```
//密码验证
int Pass_Word(){
    int password;
    do{
        printf("请输入密码\n");
        scanf("%d",&password);
        if(PASSWORD==password){
            printf("*****\n");
            printf("    欢迎进入账目管理系统\n");
            printf("*****\n");
            return 1;
        }
        else{
            printf("密码输入有误,请重新输入\n");
            printf("是否重新输入\n");
        }
    }while(getchar()=='y'||getchar()=='Y');
    return 0;
}
```

## 五、代码测试

1、当程序运行时，主菜单显示如下：

\*\*\*\*\*

餐厅服务系统

\*\*\*\*\*

- 1.点菜
  - 2.结账
  - 3.账目管理
  - 4.餐厅统计
  - 5.退出系统
- 请选择

2、此时选择相应选项会执行相应的操作，例如选择1，回车：

请选择：1

请输入来客数量:2

您和您的朋友可以到3号餐桌就餐

\*\*\*\*\*本店菜谱\*\*\*\*\*

1	葱花炒鸡蛋	4.00
2	红烧鲤鱼	66.00
3	麻辣豆腐	4.00
4	东坡肘子	33.00
5	北京烤鸭	108.00
6	鲤鱼跃龙门	666.00
7	南京叫化鸡	77.00
8	白菜炖粉条	10.00
9	四喜丸子	88.00
10	羊肉泡馍	20.00

请您点菜并输入相应的编号,输入0结束:1

请您点菜并输入相应的编号,输入0结束:2

请您点菜并输入相应的编号,输入 0 结束:3

请您点菜并输入相应的编号,输入 0 结束:0

### 3、结账:

请选择:2

请输入您就餐的餐桌编号:3

您点菜的情况如下:

编号	菜名	价格
1	葱花炒鸡蛋	4.00
2	红烧鲤鱼	66.00
3	麻辣豆腐	4.00

您一共点了 3 道菜

您本次的消费额为 74.00 元

谢谢您的惠顾,欢迎下次光临

账目保存成功

历史记录保存成功

### 4、账目管理:

请选择: 3

\*\*\*\*\*

欢迎进入账目管理系统

\*\*\*\*\*

1. 修改密码

2. 查看账本

3. 退出

请选择: 2

消费金额	消费时间
------	------

37.000000	Fri May 02 14:52:49 2008
-----------	--------------------------

37.000000	Fri May 02 14:53:05 2008
-----------	--------------------------

4.000000	Fri May 02 15:01:05 2008
----------	--------------------------

4.000000	Fri May 02 15:06:38 2008
----------	--------------------------

4.000000	Fri May 02 15:23:23 2008
----------	--------------------------

70.000000	Fri May 02 15:27:56 2008
-----------	--------------------------

### 5、餐厅统计:

请选择: 4

比较受欢迎的四道菜是:

红烧鲤鱼

麻辣豆腐

葱花炒鸡蛋

东坡肘子

## 六、思考题

- 1、怎样用多个.c 文件来实现本程序?
- 2、本例中对文件的操作都是对文本文件的操作, 请思考怎样用二进制文件进行信息的存储与读取? 有何区别?
- 3、本例中餐桌的初始化是手工的初始化, 怎样用文件代替手工初始化?
- 4、本例中没有客户换菜的功能模块, 如何完善?

## 第四章 职工信息管理

本章要求设计并实现一个职工信息管理程序。要求利用结构体存储每个职工的信息，并使用结构体数组来存储全部的职工信息。能够实现对职工信息的添加、删除、查询、修改，新建列表，排序等功能；每个功能需要细分为多个小模块，以便实现更加详尽的功能；可以将数组中的信息保存到指定文件中，并且可以对指定文件进行读取以将信息读入到数组中，也可将多个文件组成一个文件进行管理。

### 一、实践目的

- 1、巩固 C 语言的基本语法，掌握函数设计方法和结构化设计思想，
- 2、掌握结构体数组的定义和使用。
- 3、掌握用 C 语言对文件进行读写操作。掌握将结构体数据写入文件和从文件中读取结构体数据的方法。
- 4、了解规范的程序设计思想和程序编码格式。
- 5、了解职工管理过程中所需要处理的信息以及相关的处理方法。

### 二、基本要求

- 1、职工信息包括编号、姓名、性别、婚姻状况、学位、所属部门、职位、工资、工龄等信息，定义结构体储存职工信息。
- 2、用数组临时保存输入的职工信息，并且可以对现有列表中的职工信息进行增加，修改，查询，删除等基本操作。
- 3、使用二进制文件完成职工信息的存储与读取，要求每次运行某个操作时可以将文件数据读入结构体中，并给用户提供保存选项，可以将结构体中的完成操作后的数据保存在二进制文件中。
- 4、系统以菜单方式工作，制作完成后应实现类似图 4.1 所示界面。

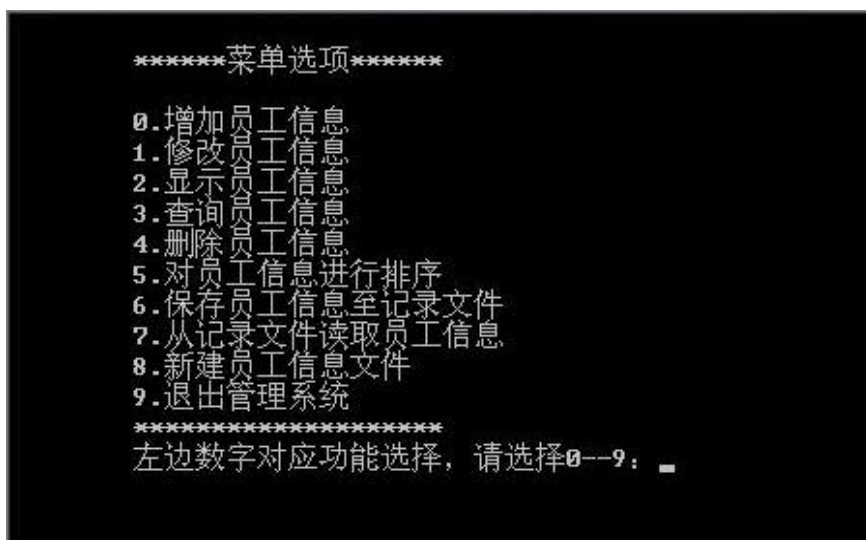


图 4.1 程序运行界面

### 三、算法分析

#### 1、数据结构

(1) 设计一个存储一条职工信息的结构体，可参考使用下述的结构体数据类型定义：

```
struct Employee
{
    char EmployeeNum[15];           /*员工的编号*/
    char EmployName[20];            /*员工的姓名*/
    char EmployGender[20];          /*员工性别*/
    char EmployeeMarriage[20];      /*员工婚姻状况*/
    char EmployeeDegree[20];        /*员工学位*/
    char EmployeeDepart[20];        /*员工所属部门*/
    char EmployeePosition[20];      /*员工职位*/
    char EmployeePay[20];           /*员工工资*/
    char EmployeeWorkTime[20];      /*员工工龄*/
};
typedef struct Employee EmpInfo;
```

(2) 在头文件中采用了宏定义，用整型、符点型、字符型等来定义所包含的数据和变量；在头文件中定义全局变量，并统一了名称和数值。

#### 2、函数定义

程序代码由五个源文件和一个头文件构成，具体文件内容和函数功能说明如表 4.1 所示。

表 4.1 文件及函数说明

源文件/头文件	函数原型	功能说明
main_menu_sel_new_quit.c	void main(void)	总控整个程序
	int menu_select(void)	显示界面，接收用户选择的命令代码
	void handle_menu(void)	处理用户选择的操作命令
	void newRecords(void)	重新在内存中建立职工信息列表
	void quit(void)	结束程序运行
dis_add.c	void addRecord(void)	增加记录
	void display(void)	显示内存里的记录员工信息
	void addInfo(int num)	写入指定员工信息
	void printInfo(int num)	打印指定员工信息
find_del_modify.c	void deleteRecord(void)	删除内存中所选定的操作记录
	int findRecord(char* target,int targetType,int from)	查找指定的记录
	void findInfo(void)	查询内存中指定的员工的信息
	void modifyRecord(void)	修改内存中指定员工的信息
save_load.c	int saveRecords(void)	将记录存入指定文件
	int loadRecords(void)	将指定文件里的记录文件读入内存
sort_copy.c	void sortInfo(void)	对内存中的员工信息按要求进行排序
	void copyRecord(EmpInfo* src,EmpInfo* temp)	将 src 指向的记录复制给 temp 指向的记录



employee.h	静态变量及常数定义	提供静态变量及常数
	结构体定义	员工信息结构
	库函数及自定义函数声明	调用库函数及自定义函数

### 3、处理过程

(1) 为一些函数设计一个整型返回值用来区分函数操作是否正确。每个员工的信息用一个 **EmplInfo** 的结构体来保存，用一个 **EmplInfo** 类型的数组 **records** 来保存一组员工信息。用宏定义 **INITIAL\_SIZE** 来规定这个数组的大小，如果所定的数组大小不够时，用另一个宏定义 **INCR\_SIZE** 来动态申请内存增加数组大小。用全局变量 **numEmpl** 来记录学生数，**arraySize** 为数组分配空间大小。

(2) **saveFlag** 是员工信息是否保存的标志，当员工信息已经保存到文件就设置为 0 表示“已保存”，如果没有存入文件就设置为 1 表示“为保存”，以此为关键点展开对文件的保存，载入，叠加等相关操作。

#### (3) 主要函数的算法

##### 1) addRecord 函数

用来在当前链表尾部增加新的信息，只要将新的信息保存到 **record[numEmpl]**，中即可，然后 **numEmpl** 自加 1，完成操作。如果再增加新的信息前，**numEmpl** 已经大于或等于 **arraySize**，这时就要使用 **realloc** 函数重新分配一块大小为 (**arraySize+INCR\_SIZE**) **EmplInfo** 的数组的存储块，并重新设置 **arraySize**。

##### 2) loadRecord 函数

进行两个文件拼接时，在内存无记录的情况下，对读入的第二个文件进行询问。如果覆盖原来记录，就先保存原纪录，然后令 **numEmpl=0**。否则原来的 **numEmpl** 不变。在读取文件时，使用 **fread** 函数，每次读取 **sizeof(EmplInfo)** 个字节，存在数组 **records[numEmpl]** 中，并令 **numEmpl** 自加一，如此下去，直到读完文件。如果在每读入一条信息之前，**numEmpl** 已经大于或等于 **arraySize**，那么就使用 **realloc** 函数重新为 **records** 分配大小为 (**arraySize+INCR\_SIZE**) **EmplInfo** 的数组的存储块，并重新设置 **arraySize**。

##### 3) deleteRecord, findInfo, modifyInfo 函数

这三个函数有一个共同之处，都是调用 **findRecord** 函数从指定序号开始，顺序查找指定条件的记录，若要找到就返回指定记录的序号并且显示指定员工信息，没找到就返回 -1。**deleteRecord** 删除记录操作函数，让用户选择是否确定删除记录，若决定执行删除操作，则 **numEmpl** 自减 1，然后将数组排在删除记录后的记录往前挪一格。**modifyInfo** 修改记录操作函数，让用户选择是否确定修改记录，若决定执行修改操作，则重新输入指定员工信息。

##### 4) sortInfo 函数

此排序操作函数主要应用冒泡排序法进行排序。

## 四、参考代码

```
"Employee.h"
#ifndef H_STUDENT_HH
#define H_STUDENT_HH
#include<stdio.h>
```

```

#include<stdlib.h>
#include<conio.h>
#include<string.h>
#define INITIAL_SIZE 100 /*数组初始大小*/
#define INCR_SIZE 50 /*数组每次增加的大小*/
struct Employee {
    char EmployeeNum[15]; /*员工的编号*/
    char EmployName[20]; /*员工的姓名*/
    char EmployGender[20]; /*员工性别*/
    char EmployeeMarriage[20]; /*员工婚姻状况*/
    char EmployeeDegree[20]; /*员工学位*/
    char EmployeeDepart[20]; /*员工所属部门*/
    char EmployeePosition[20]; /*员工职位*/
    char EmployeePay[20]; /*员工工资*/
    char EmployeeWorkTime[20]; /*员工工龄*/
};
typedef struct Employee EmpInfo;
extern int numEmpl; /*记录的员工数*/
extern EmpInfo*records; /*记录员工信息的数组*/
extern char savedFlag; /*信息是否已保存的标志,0 为已保存,1
为未保存*/
extern int arraySize; /*数组大小*/
void handle_menu(void);
int menu_select(void);
void addRecord(void);
void modifyRecord(void);
void display(void);
void findInfo(void);
void deleteRecord(void);
void sortInfo(void);
int saveRecords(void);
int loadRecords(void);
void newRecords(void);
void quit(void);
int findRecord(char* target,int targetType,int from);
void copyRecord(EmpInfo* src,EmpInfo* dest);
void addInfo(int num);
void printInfo(int num);
#endif // H_STUDENT_HH

"main_menu_sel_new_quit.c"
#include "employee.h"
/*****
* 初始化
*****/
int numEmpl=0; /*记录的员工数*/
EmpInfo*records=NULL; /*记录员工信息的数组*/
char savedFlag=0; /*信息是否已保存的标志, 1 为未保存, 0 为已保
存*/
int arraySize; /*数组大小*/
/*****
* 主函数
*****/
void main(void){
    printf("\n");
    printf("\t*****\n");
    printf("\t*软件说明: *");
    printf("\t* 本软件用于员工信息管理, 是您的理想助手 。 *");

```

```

printf("\t*                                *\n");
printf("\t*      欢迎您使用本管理系统!      *\n");
printf("\t*                                *\n");
printf("\t*              LJM V3.0   测试版      *\n");
printf("\t*      涅磐++蜕变工作室为您呈现      *\n");
printf("\t*****\n");
printf("\n");
getch();
system("cls");
/*初始化数组*/
records=(EmpInfo*)malloc(sizeof(EmpInfo)*INITIAL_SIZE);
if(records==NULL){
    printf("memory fail!");
    exit(-1);
}
arraySize=INITIAL_SIZE;
handle_menu();
}
/******
*      菜单处理函数
*****/
void handle_menu(void){
    for(;;){
        switch(menu_select()){
            case 0:
                addRecord();
                getch();
                system("cls");          //清屏
                break;
            case 1:
                modifyRecord();
                system("cls");
                break;
            case 2:
                display();
                getch();
                system("cls");
                break;
            case 3:
                findInfo();
                system("cls");
                break;
            case 4:
                deleteRecord();
                system("cls");
                break;
            case 5:
                sortInfo();
                system("cls");
                break;
            case 6:
                saveRecords();
                system("cls");
                break;
            case 7:
                loadRecords();
                getch();
                system("cls");

```

```

        break;
    case 8:
        newRecords();
        getch();
        system("cls");
        break;
    case 9:
        quit();
    }
}
}
/*****
*   菜单选择函数 menu_selected
*****/
int menu_select(void){
    char s[2];
    int choice=0;
    printf("\n");
    printf("\t*****菜单选项*****\n");
    printf("\n");
    printf("\t0.增加员工信息\n");
    printf("\t1.修改员工信息\n");
    printf("\t2.显示员工信息\n");
    printf("\t3.查询员工信息\n");
    printf("\t4.删除员工信息\n");
    printf("\t5.对员工信息进行排序\n");
    printf("\t6.保存员工信息至记录文件\n");
    printf("\t7.从记录文件读取员工信息\n");
    printf("\t8.新建员工信息文件\n");
    printf("\t9.退出管理系统\n");
    printf("\t*****");
    printf("\n\t 左边数字对应功能选择，请选择 0--9: ");
    for(; ){
        scanf("%s",s);
        choice=atoi(s);          /*处理键入的非数字键，过滤出数字 0*/
        if(choice==0&&(strcmp(s,"0")!=0))
            choice=11;
        if(choice<0||choice>9)
            printf("\n\t 输入错误，重选 0--9: ");
        else
            break;
    }
    return choice;
}
/*****
*   新建员工信息记录
*   结果:若原来信息没有保存,则保存原来的信息,然后重新输入信息记录
*****/
void newRecords(void){
    char str[5];
    if(numEmpl!=0) {
        if(savedFlag==1){
            printf("现在已经有记录，选择处理已有记录的方法。 \n");
            printf("是否保存原来的记录? (y/n)");
            scanf("%s",str);
            if(str[0]!='n' && str[0]!='N')
                saveRecords();
        }
    }
}

```

```

    }
    numEmpl=0;
    addRecord();
}
/*****
*      结束运行，退出
*      参数 void
*****/
void quit(void){
    char str[5];
    if(savedFlag==1) {
        printf("是否保存原来的记录? (y/n)");
        scanf("%s",str);
        if(str[0]!='n' && str[0]!='N')
            saveRecords();
    }
    free(records);
    exit(0);
}
"dis_add.c"
#include "employee.h"
/*****
*      显示所有的员工信息
*****/
void display(void){
    int i;
    char str[3];
    if(numEmpl==0) {
        printf("没有可供显示的记录!");
        return;
    }
    printf("\n1.显示简明员工信息\n2.显示详细员工信息\n 您的选择:");
    scanf("%s",str);
    if(str[0]=='2'){
        for(i=0;i<numEmpl;i++){
            /*打印员工信息*/
            printInfo(i);
            /*打印满 20 个记录后停下来*/
            if(i%20==0 && i!=0){
                printf("输入任意字符后继续...");
                getch();
                printf("\n\n");
            }
        }
    }
    if(str[0]=='1'){
        for(i=0;i<numEmpl;i++){
            printf("编号\t姓名\t部门\t职位\n");
            printf("%s\t%s\t%s\t%s\n",records[i].EmployeeNum,records[i].EmployeeName,records[i].EmployeeDepart,records[i].EmployeePosition);
            if(i%20==0 && i!=0){
                printf("输入任意字符后继续...");
                getch();
                printf("\n\n");
            }
        }
    }
}
}

```

```

/*****
*      在当前表的末尾增加新的信息
*      结果: records 中将记录新的信
*      息, 如果数组大小不够, 会重
*      新申请数组空间
*****/
void addRecord(void){
    char str[10];
    if(numEmpl==0)
        printf("原来没有记录, 现在建立的记录\n");
    else
        printf("下面在当前表的末尾增加新的信息\n");
    while(1){
        printf("你将要添加一组员工信息, 确定吗? (y/n)");
        scanf("%s",str);
        if(str[0]=='n' || str[0]=='N')                /*不再添加新的信息*/
            break;
        if(numEmpl>=arraySize){                      /*数组空间不足, 需要重新申请空间*/
            records=realloc(records,(arraySize+INCR_SIZE)*sizeof(EmpInfo));
            if(records==NULL){
                printf("memory failed!");
                exit(-1);
            }
            arraySize=arraySize+INCR_SIZE;
        }
        addInfo(numEmpl);
        numEmpl++;
    }
    printf("现在一共有%d 条信息\n",numEmpl);
    savedFlag=1;
}
/*****
*      输入指定员工信息
*****/
void addInfo(int num){
    char str1[10];
    char str2[10];
    char str3[10];
    char str4[10];
    printf("请输入员工编号: ");                      /*员工信息输入*/
    scanf("%s",records[ num].EmployeeNum);
    printf("请输入员工姓名: ");
    scanf("%s",records[num].EmployName);
    printf("请输入员工性别 (0 为女, 1 为男): ");
    scanf("%s",str1);
    if(str1[0]=='0')
        strcpy(records[num].EmployGender,"女");
    else
        strcpy(records[num].EmployGender,"男");
    printf("请输入员工婚姻状况 (0 为未婚, 1 为已婚): ");
    scanf("%s",str2);
    if(str2[0]=='0')
        strcpy(records[num].EmployeeMarriage,"未婚");
    else
        strcpy(records[num].EmployeeMarriage,"已婚");
    printf("请输入员工学历: (0.大学本科生, 1.硕士研究生, 2.高等博士生:");
    scanf("%s",str3);
    if(str3[0]=='0')
        strcpy(records[num].EmployeeDegree,"大学本科生");

```

```

else if(str3[0]=='1')
    strcpy(records[num].EmployeeDegree,"硕士研究生");
else if(str3[0]=='2')
    strcpy(records[num].EmployeeDegree,"高等博士生");
printf("请输入员工所在部门: ");
scanf("%s",records[num].EmployeeDepart);
printf("请输入员工职务: (0.职员, 1.主任, 2.经理, 3.董事长)");
scanf("%s",str4);
if(str4[0]=='0')
    strcpy(records[num].EmployeePosition,"职员");
if(str4[0]=='1')
    strcpy(records[num].EmployeePosition,"主任");
if(str4[0]=='2')
    strcpy(records[num].EmployeePosition,"经理");
if(str4[0]=='3')
    strcpy(records[num].EmployeePosition,"董事长");
printf("请输入员工工资: ");
scanf("%s",records[num].EmployeePay);
printf("请输入员工工龄: (以年为单位计算):");
scanf("%s",records[num].EmployeeWorkTime);
}
/*****
*      打印指定员工信息
*****/
void printInfo(int num){
    printf("编号\t姓名\t性别\t婚姻状况\t学位\t部门\t职位\t工资\t工龄\n");
    printf("%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n",records[num].EmployeeNum,records[num].
EmployeeName,records[num].EmployGender,records[num].EmployeeMarriage,records[num].Empl
oyeeDegree,records[num].EmployeeDepart,records[num].EmployeePosition,records[num].
EmployeePay,records[num].EmployeeWorkTime);
}
"find_del_modify.c"
#include "employee.h"
/*****
*      查找指定的记录
*      参数: target:欲查找记录的某一项与 target 相同
*      targetType:表明通过那一项来查找, 0 为学号, 1 为姓名, 2 为名次
*      from:从第 from 个记录开始找
*      返回: 找到的记录的序号, 若找不到则返回-1
*****/
int findRecord(char* target,int targetType,int from){
    int i;
    for(i=from;i<numEmpl;i++){
        if((targetType==0 && strcmp(target,records[i].EmployeeNum)==0)||
(targetType==1 && strcmp(target,records[i].EmployName)==0)||
(targetType==2 && strcmp(target,records[i].EmployeeDepart)==0))
            return i;
    }
    return -1;
}
/*****
*      查找指定员工的信息
*      可以按照员工编号, 姓名, 所在部门进行查找
*****/
void findInfo(void){
    char str[5];
    char target[20];
    int type;

```

```

int count;
int i;
if(numEmpl==0) {
    printf("没有可供查询的记录! ");
    getch();
    return;
}
//输出查询菜单,并处理输入信息
while(1){
    printf("请输入查找的方式: \n");
    printf("1.按员工编号\n");
    printf("2.按员工姓名\n");
    printf("3.按员工所在部门\n");
    printf("4.结束查询\n");
    scanf("%s",str);
    if(str[0]=='4')
        return;
    if(str[0]=='1'){
        printf("请输入欲查询的员工的编号: ");
        scanf("%s",target);
        type=0;
    }
    else if(str[0]=='2'){
        printf("请输入欲查找的员工的姓名: ");
        scanf("%s",target);
        type=1;
    }
    else if(str[0]=='3'){
        printf("请输入欲查找的员工所在部门: ");
        scanf("%s",target);
        type=2;
    }
    i=findRecord(target,type,0);
    if(i!=-1){
        printf("编号\t姓名\t性别\t婚姻状况\t学位\t部门\t职位\t工资\t工龄\n");
    }
    count=0;
    while(i!=-1){
        printf("%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n",records[i].EmployeeNum,records[i].EmployeeName,records[i].EmployeeGender,records[i].EmployeeMarriage,records[i].EmployeeDegree,records[i].EmployeeDepart,records[i].EmployeePosition,records[i].EmployeePay,records[i].EmployeeWorkTime);
        i=findRecord(target,type,i+1);
        count++;
    }
    if(count==0)
        printf("没有符合条件的员工! \n");
    else
        printf("一共找到了%d 名员工的信息\n\n",count);
}
}
/*****
*      删除指定的记录
*****/
void deleteRecord(void){
    char str[5];
    char target[20];
    int type;

```



```

int i,j;
if(numEmpl==0) {
    printf("没有可供删除的记录! ");
    getch();
    return;
}
while(1){
    printf("请输入如何找到欲删除的记录的方式: \n");
    printf("1.按员工编号\n");
    printf("2.按员工姓名\n");
    printf("3.退出删除操作\n");
    scanf("%s",str);
    if(str[0]=='3')
        return;
    if(str[0]=='1'){
        printf("请输入该员工编号: ");
        scanf("%s",target);
        type=0;
    }
    else if(str[0]=='2'){
        printf("请输入该员工的姓名: ");
        scanf("%s",target);
        type=1;
    }
    i=findRecord(target,type,0);
    if(i==-1)
        printf("没有符合条件的员工! \n");
    while(i!=-1){
        /*打印员工信息*/
        printInfo(i);
        printf("确定要删除这个员工信息吗? (y/n)");
        scanf("%s",str);
        if(str[0]=='y'||str[0]=='Y'){
            numEmpl--;
            /*将后面的记录前移*/
            for(j=i;j<numEmpl;j++){
                copyRecord(&records[j+1],&records[j]);
            }
            printf("成功删除纪录!\n");
        }
        else
            break;
        /*取下一个符合条件的记录*/
        i=findRecord(target,type,i++);
    }
}
savedFlag=1;
}
/*****
*      修改指定员工的信息
*****/
void modifyRecord(void){
    char str[5];
    char target[10];
    int type;
    int i;
    if(numEmpl==0){
        printf("没有可供修改的记录! ");

```

```

        getch();
        return;
    }
    while(1){
        printf("请输入如何找到欲修改的记录的方式: \n");
        printf("1.按员工编号\n");
        printf("2.按员工姓名\n");
        printf("3.退出修改操作\n");
        scanf("%s",str);
        if(str[0]=='3')
            return;
        if(str[0]=='1'){
            printf("请输入该员工编号: ");
            scanf("%s",target);
            type=0;
        }
        if(str[0]=='2'){
            printf("请输入该员工姓名: ");
            scanf("%s",target);
            type=1;
        }
        i=findRecord(target,type,0);
        if(i==-1)
            printf("没有符合条件的员工! \n");
        while(i!=-1){
            /*打印员工信息*/
            printInfo(i);
            printf("确定要修改这个员工的信息吗? (y/n)");
            scanf("%s",str);
            if(str[0]=='y' || str[0]=='Y'){
                addInfo(i);
                printf("修改记录成功! \n");
            }
            i=findRecord(target,type,i+1);
        }
    }
    savedFlag=1;
}

"save_load.c"
#include "employee.h"
/*****
*    文件储存操作函数
*    结果: 数组 records 被保存至指定文件
*    返回: 成功 0, 失败-1
*****/
int saveRecords(void){
    FILE *fp;
    char fname[30];
    if(numEmpl==0)    {
        printf("\n\n 没有记录可存! \n\n");
        getch();
        return -1;
    }
    printf("\n\n 输入要存入的文件名:");
    scanf("%s",fname);
    if((fp=fopen(fname,"wb"))==NULL){
        printf("不能存入文件! \n");
        getch();
    }
}

```

```

        return -1;
    }
    printf("\n 存入文件...\n");
    fwrite(records, sizeof(EmpInfo) * numEmpl, 1, fp);
    fclose(fp);
    printf("%d 条记录已经存入文件，请继续操作。 \n", numEmpl);
    savedFlag = 0; /*更新是否已保存的标志*/
    getch();
    return 0;
}
/*****
*      文件读取操作函数
*      结果： records 将为从指定文件中读取出的记录
*      返回： 成功 0， 失败 -1
*****/
int loadRecords(void){
    FILE *fp;
    char fname[30];
    char str[5];
    if(numEmpl != 0 && savedFlag == 0){
        printf("请选择您是要覆盖现有记录(y)，还是要将");
        printf("读取的记录添加到现有记录之后(n)? \n");
        scanf("%s", str);
        if(str[0] == 'n' || str[0] == 'N'){
            /*将读取的记录添加到现有记录之后*/
            savedFlag = 1;
        }
        else{
            if(savedFlag == 1){
                /*覆盖现有记录*/
                scanf("%s", str);
                if(str[0] != 'n' && str[0] != 'N')
                    printf("读取文件将会更改原来的记录，");
                printf("是否保存原来的记录？ (Y/n) ");
                saveRecords();
            }
            numEmpl = 0;
        }
    }
    printf("\n\n 请输入要读取的文件名:");
    scanf("%s", fname);
    if((fp = fopen(fname, "rb")) == NULL){
        printf("\n\n 打不开文件！ 请重新选择\n\n");
        return -1;
    }
    printf("\n 取文件...\n");
    while(!feof(fp)){
        /*现在的数组空间不足，需要重新申请空间*/
        if(numEmpl >= arraySize){
            records = realloc(records, (arraySize + INCR_SIZE) * sizeof(EmpInfo));
            if(records == NULL){
                printf("memory failed!");
                exit(-1);
            }
            arraySize = arraySize + INCR_SIZE;
        }
        if(fread(&records[numEmpl], sizeof(EmpInfo), 1, fp) != 1)
            break;
    }
}

```

```

        numEmpl++;
    }
    fclose(fp);
    printf("现在共有%d 条记录。",numEmpl);
    return 0;
}
"sort_copy.c"
#include "employee.h"
/*****
*   文件排序操作函数
*   返回: void
*****/
void sortInfo(void){
    char str[5];
    int i,j;
    EmpInfo temp;
    if(numEmpl==0){
        printf("没有可供排序的记录! ");
        getch();
        return;
    }
    printf("请输入您希望进行的排序的方式: \n");
    printf("1.按编号进行排序\n");
    printf("2.按部门进行排序\n");
    printf("3.按工资进行排序\n");
    printf("4.按工龄进行排序\n");
    printf("5.按错了, 我不想进行排序\n");
    scanf("%s",str);
    if(str[0]<'1' || str[0]>'4')
        return;
    /*进行排序*/
    for(i=0;i<numEmpl-1;i++){
        for(j=i+1;j<numEmpl;j++){
            if((str[0]=='1'&&strcmp(records[i].EmployeeNum,
                records[j].EmployeeNum)>0)||
                (str[0]=='2'&&strcmp(records[i].EmployeeDepart,
                records[j].EmployeeDepart)<0)||
                (str[0]=='3'&&
                strcmp(records[i].EmployeePay,records[j].EmployeePay)>0)||
                (str[0]=='4'&&
                strcmp(records[i].EmployeeWorkTime,records[j].EmployeeWorkTime)>0)){
                copyRecord(&records[i],&temp);
                copyRecord(&records[j],&records[i]);
                copyRecord(&temp,&records[j]);
            }
        }
    }
    printf("排序已经完成\n");
    getch();
    savedFlag=1;
}
/*****
*   将 src 指向的一条记录复制给 temp 指向的记录
*****/
void copyRecord(EmpInfo* src,EmpInfo* temp){
    strcpy(temp->EmployeeNum,src->EmployeeNum);
    strcpy(temp->EmployName,src->EmployName);
    strcpy(temp->EmployGender,src->EmployGender);

```

```

strcpy(temp->EmployeeMarriage,src->EmployeeMarriage);
strcpy(temp->EmployeeDegree,src->EmployeeDegree);
strcpy(temp->EmployeeDepart,src->EmployeeDepart);
strcpy(temp->EmployeePosition,src->EmployeePosition);
strcpy(temp->EmployeePay,src->EmployeePay);
strcpy(temp->EmployeeWorkTime,src->EmployeeWorkTime);
}

```

## 五、代码测试

1、当程序运行时，主菜单显示如下，以后每一项操作完后都会打印此菜单：

\*\*\*\*\*菜单选项\*\*\*\*\*

- 0.增加员工信息
- 1.修改员工信息
- 2.显示员工信息
- 3.查询员工信息
- 4.删除员工信息
- 5.对员工信息进行排序
- 6.保存员工信息至记录文件
- 7.从记录文件读取员工信息
- 8.新建员工信息文件
- 9.退出管理系统

\*\*\*\*\*

左边数字对应功能选择，请选择 0--9：

分别选择 1——6 操作，分别得到如下信息：

没有可供修改的记录！

没有可供显示的记录！

没有可供查询的记录！

没有可供删除的记录！

没有可供排序的记录！

没有记录可存！

选择 0 则输出：原来没有记录，现在建立的记录

你将要添加一组员工信息，确定吗？(y/n)

选择 7 输出：请输入要读取的文件名:nothing

打不开文件！请重新选择

选择 8 输出：原来没有记录，现在建立的记录

你将要添加一组员工信息，确定吗？(y/n)

选择 9 输出：Press any key to continue

选择 0——9 以外的数字输出：输入错误，重选 1——9：

2、选择“0”员工信息输入函数测试：

输出：

原来没有记录，现在建立的记录

你将要添加一组员工信息，确定吗？(y/n)y

请输入员工编号：1

请输入员工姓名：hangeng

请输入员工性别（0 为女，1 为男）：0

请输入员工婚姻状况 (0 为未婚, 1 为已婚): 1  
 请输入员工学历: (0.大学本科生, 1.硕士研究生, 2.高等博士生):0  
 请输入员工所在部门: management  
 请输入员工职务: (0.职员, 1.主任, 2.经理, 3.董事长)2  
 请输入员工工资: 1345  
 请输入员工工龄: (以年为单位计算):12  
 你将要添加一组员工信息, 确定吗? (y/n)y  
 请输入员工编号: 2  
 请输入员工姓名: wanghai  
 请输入员工性别 (0 为女, 1 为男): 0  
 请输入员工婚姻状况 (0 为未婚, 1 为已婚): 1  
 请输入员工学历: (0.大学本科生, 1.硕士研究生, 2.高等博士生):2  
 请输入员工所在部门: basement  
 请输入员工职务: (0.职员, 1.主任, 2.经理, 3.董事长)1  
 请输入员工工资: 2341  
 请输入员工工龄: (以年为单位计算):11  
 你将要添加一组员工信息, 确定吗? (y/n)n  
 现在一共有 2 条信息

### 3、选择“2”显示信息函数测试:

输出:

- 1.显示简明员工信息
- 2.显示详细员工信息

您的选择:1

编号	姓名	部门	职位
1	hangeng	management	经理
2	wanghai	basement	主任

### 4、选择“1”修改信息函数测试:

输出:

请输入如何找到欲修改的记录的方式:

- 1.按员工编号
- 2.按员工姓名
- 3.退出修改操作

1

请输入该员工编号: 1

编号	姓名	性别	婚姻状况	学位	部门	职位	工资
1	hangeng	女	已婚	大学本科生	management	经理	2356

确定要修改这个员工的信息吗? (y/n)y

请输入员工编号: 5

请输入员工姓名: hanggeng

请输入员工性别 (0 为女, 1 为男): 0

请输入员工婚姻状况 (0 为未婚, 1 为已婚): 1

请输入员工学历: (0.大学本科生, 1.硕士研究生, 2.高等博士生):2

请输入员工所在部门: management

请输入员工职务: (0.职员, 1.主任, 2.经理, 3.董事长)3

请输入员工工资: 2353

请输入员工工龄: (以年为单位计算):13

修改记录成功!

5、选择“3”查询信息函数测试:

输出:

请输入查找的方式:

- 1.按员工编号
- 2.按员工姓名
- 3.按员工所在部门
- 4.结束查询

3

请输入欲查找的员工所在部门: management

编号	姓名	性别	婚姻状况	学位	部门	职位	工资	工龄
5	hanggeng	女	已婚	高等博士生	management	董事长	2353	13

一共找到了 1 名员工的信息

6、选择“4”删除员工信息函数测试:

输出:

请输入如何找到欲删除的记录的方式:

- 1.按员工编号
- 2.按员工姓名
- 3.退出删除操作

1

请输入该员工编号: 2

编号	姓名	性别	婚姻状况	学位	部门	职位	工资	工龄
2	wanghai	女	已婚	高等博士生	basement	主任	2341	11

确定要删除这个员工信息吗? (y/n)y

成功删除纪录!

7、选择“5”测试员工信息排序函数:

建议输入多组不同员工信息争取做到各个信息都有差别, 然后执行该函数。操作完成后选择“2”显示员工信息测试排序是否正确。

8、选择“6”保存文件函数测试:

输出:

输入要存入的文件名:employee

存入文件...

1条记录已经存入文件, 请继续操作。

9、选择“7”载入文件函数测试:

输出:

请选择您是要覆盖现有记录(y), 还是要将读取的记录添加到现有记录之后(n)?

y

请输入要读取的文件名:employee

取文件...

现在共有 1 条记录。

10、选择“8”新建员工信息函数测试:

输出:

原来没有记录, 现在建立的记录

你将要添加一组员工信息, 确定吗? (y/n)y

请输入员工编号: 5

请输入员工姓名: meili

请输入员工性别 (0 为女, 1 为男): 0

请输入员工婚姻状况 (0 为未婚, 1 为已婚): 1

请输入员工学历: (0.大学本科生, 1.硕士研究生, 2.高等博士生):2

请输入员工所在部门: management

请输入员工职务: (0.职员, 1.主任, 2.经理, 3.董事长)1

请输入员工工资: 2365

请输入员工工龄: (以年为单位计算):2

你将要添加一组员工信息, 确定吗? (y/n)n

现在一共有 1 条信息

*11、选择“9”退出程序操作函数:*

输出:

是否保存原来的记录? (y/n)

y

输入要存入的文件名:employee

存入文件...

1 条记录已经存入文件, 请继续操作。

Press any key to continue

## 六、思考题

- 1、程序中的结构数组是如何工作的? 怎样对结构数组动态分配内存空间?
- 2、头文件中使用的预处理命令有什么作用? 这样做有什么好处?
- 3、经过本次程序实践你认为 C 语言程序从开发到执行一般需要几个阶段? 各个阶段的作用是什么?



## 第五章 设备信息管理

本章要求设计并实现一个设备信息管理程序。要求实现设备入库登记、设备领用与归还登记、设备维修登记、设备变更登记等功能模块，并且每一个模块还需要划分多个子模块；使用各自不同的结构体来储存不同的登记信息；使用链表来实现各种登记信息的添加、删除、查询、修改等操作；使用文件来保存数据，以便下次运行时可以自动将数据从文件读取到链表中。

### 一、实践目的

- 1、掌握带头节点的链表的工作原理和处理方式。
- 2、掌握使用 `malloc`、`free` 等函数对链表进行创建、增加、删除等操作的方法，掌握遍历链表以实现查询等操作的方法。
- 3、加深理解模块化的编程思想，将一个程序划分成不同的源文件、函数来编写，掌握对多个源文件进行处理的方式。
- 4、掌握使用 C 语言对文件进行读写操作，掌握将结构体数据直接写入文件和从文件中读取结构体数据的方法。
- 5、了解设备管理过程中所需要处理的信息以及相关的处理方法。

### 二、基本要求

- 1、本程序要求实现设备入库登记、设备领用与归还登记、设备维修登记、设备变更登记四大功能模块，每个功能模块又分别对应登记信息的显示、查询、增加、删除和修改等操作子模块，从而完成一个设备信息管理系统所需功能。
- 2、要求使用四种不同的结构体来分别存储四大功能模块中所需登记的信息。
- 3、使用链表来实现对登记信息的显示、查询、增加、删除等操作。
- 4、使用二进制文件完成数据的存取，要求每次运行某个功能模块时将数据读入结构体中，并给用户提供保存选项，可以将结构体中的数据保存在二进制文件中。
- 5、系统制作完成后应实现类似图 5.1 所示界面。

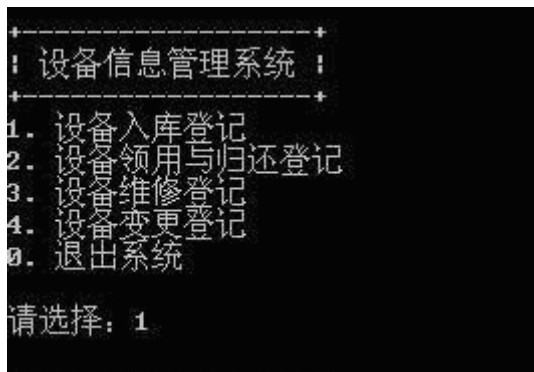


图 5.1 程序运行界面

### 三、算法分析

#### 1、数据结构

程序需要设计 4 个结构体来满足存储不同类型登记数据的需要，结构体设计参考如下：

```
struct ent_node    /*存储入库登记的结构体*/
{
    char id[10];
    char type[10];
    char brand[16];
    char model[16];
    long value;
    char buy_date[10];
    ent_ptr next;
};
struct bor_node    /*存储领用与归还登记的结构体*/
{
    char id[10];
    char dev_id[10];
    int type;
    char applicant[10];
    char transactor[10];
    char date[10];
    bor_ptr next;
};
struct fix_node    /*存储维修登记的结构体*/
{
    char id[10];
    char dev_id[10];
    char reporter[10];
    char date[10];
    int type;
    fix_ptr next;
};
struct mod_node    /*存储变更登记的结构体*/
{
    char id[10];
    char dev_id[10];
    char date[10];
    char before[20];
    char after[20];
    mod_ptr next;
};
```

#### 2、函数定义

程序由 devman.c、entreg.c、borreg.c、fixreg.c、modreg.c 等 5 个源文件构成，由 entreg.h、borreg.h、fixreg.h、modreg.h 等 4 个头文件构成。其中命名为\*reg.\*的文件用于定义各模块所需结构体及相关函数，由于实现功能类似，为节省篇幅，仅以 entreg.c 及 entreg.h 举例，函数说明如表 5.1 所示。

表 5.1 文件及函数说明

源文件	函数名或其他部分	功能
devman.c	int main(int argc, char** argv)	主控制函数，处理菜单操作
	void print_main()	打印主菜单
	void print_enter()	打印入库登记菜单
	void print_borrow()	打印领用登记菜单

	void print_fix()	打印维修登记菜单
	void print_modify()	打印变更登记菜单
entreg.c	void clean_ent_list(ent_ptr list)	清空入库登记链表
	ent_ptr create_ent_list(void)	创建一个入库登记链表，如果数据文件存在，则从文件中读取数据
	ent_ptr create_ent_node(void)	创建一个入库登记节点
	void save_ent_list(ent_ptr list)	将入库登记链表保存到文件中
entreg.c	void insert_ent_node(ent_ptr list, ent_ptr node)	向链表尾部插入新的节点
	void delete_ent_node(ent_ptr list, ent_ptr node)	删除一个节点
	ent_ptr query_ent_node(ent_ptr list, char* id)	按编号查找指定节点
	void print_ent_list(ent_ptr list)	打印整个入库登记链表
	void print_ent_node(ent_ptr node)	打印指定入库登记节点
	void modify_ent_node(ent_ptr list, ent_ptr node)	修改指定节点中的内容
entreg.h	结构体定义	
	结构体指针类型定义	
	函数原型声明	

### 3、处理过程

以入库登记的处理过程为例。

(1) 创建链表，先使用 malloc 函数创建一个头节点，这个节点不保存设备信息，仅用来进行链表指示。创建头节点后再判断是否存在数据文件或数据文件是否为空。如果数据文件不存在或数据文件为空则直接返回链表的头节点，否则对文件进行读取操作，将文件内容读入链表中，最后将链表的头节点返回。

(2) 设计链表操作相关函数。如果要对链表进行插入操作，首先需要使用 create\_ent\_node 函数来创建一个节点，得到新创建节点的指针，再将节点插入到链表中。调用方法：

```
ent_node = create_ent_node();
insert_ent_node(ent_list, ent_node);
```

当对链表进行删除操作时，首先需要使用 query\_ent\_node 函数按编号查找到指定的节点，再使用 delete\_ent\_node 函数来删除。调用方法：

```
printf("请输入要删除的编号：");
scanf("%s", id);
getchar();
ent_node = query_ent_node(ent_list, id);
if(ent_node != NULL)
    delete_ent_node(ent_list, ent_node);
```

(3) 将链表保存到二进制文件中。需要注意的是，由于链表存在头节点，在保存的时候注意不要把没有存储设备信息的头节点保存到文件中，在为 pos 赋初值的时候应将初值赋为 list->next。

```
void save_ent_list(ent_ptr list){
```

```

FILE* fp;
ent_ptr pos = list->next; /*将 pos 赋值为 list->next*/
fp = fopen(ENTER_FILE, "wb");
while(pos != NULL) {
    fwrite(pos, sizeof(struct ent_node), 1, fp);
    pos = pos->next;
}
printf("\n");
printf("保存成功! \n");
fclose(fp);
}

```

#### 四、参考代码

这里示例 devman.c、entreg.c 和 entreg.h 三个文件中的代码。

```

/*devman.c*/
#include <stdio.h>
#include <stdlib.h>
#include "borreg.h"
#include "entreg.h"
#include "fixreg.h"
#include "modreg.h"
void print_main();
void print_enter();
void print_borrow();
void print_fix();
void print_modify();

int main(int argc, char** argv){
    char c;
    char id[10];
    int back;
    ent_ptr ent_list;
    bor_ptr bor_list;
    fix_ptr fix_list;
    mod_ptr mod_list;
    ent_ptr ent_node;
    bor_ptr bor_node;
    fix_ptr fix_node;
    mod_ptr mod_node;
    while (1) {
        print_main();
        c = getchar();
        getchar();/*吸收回车*/
        switch (c) {
            case '1':
                ent_list = create_ent_list();
                while (1) {
                    back = 0;
                    print_enter();
                    c = getchar();
                    getchar();
                    switch (c) {
                        case '1':
                            print_ent_list(ent_list);
                            break;
                        case '2':

```

```

        printf("请输入设备编号: ");
        scanf("%s", id);
        getchar();
        ent_node = query_ent_node(ent_list, id);
        if(ent_node != NULL)
            print_ent_node(ent_node);
        break;
    case '3':
        ent_node = create_ent_node();
        insert_ent_node(ent_list, ent_node);
        break;
    case '4':
        printf("请输入要删除的编号: ");
        scanf("%s", id);
        getchar();
        ent_node = query_ent_node(ent_list, id);
        if(ent_node != NULL)
            delete_ent_node(ent_list, ent_node);
        break;
    case '5':
        printf("请输入设备编号: ");
        scanf("%s", id);
        getchar();
        ent_node = query_ent_node(ent_list, id);
        print_ent_node(ent_node);
        modify_ent_node(ent_list, ent_node);
        break;
    case '6':
        clean_ent_list(ent_list);
        break;
    case '7':
        save_ent_list(ent_list);
        break;
    case '0':
        back = 1;
        break;
    default:
        printf("输入错误, 请重新输入! \n");
    }
    if (back)
        break;
}
break;
case '2':
    bor_list = create_bor_list();
    while (1) {
        back = 0;
        print_borrow();
        c = getchar();
        getchar();
        switch (c) {
            case '1':
                print_bor_list(bor_list);
                break;
            case '2':
                printf("请输入登记编号: ");
                scanf("%s", id);
                getchar();

```

```

        bor_node = query_bor_node(bor_list, id);
        if(bor_node != NULL)
            print_bor_node(bor_node);
        break;
    case '3':
        bor_node = create_bor_node();
        insert_bor_node(bor_list, bor_node);
        break;
    case '4':
        printf("请输入要删除的编号: ");
        scanf("%s", id);
        getchar();
        bor_node = query_bor_node(bor_list, id);
        if(bor_node != NULL)
            delete_bor_node(bor_list, bor_node);
        break;
    case '5':
        printf("请输入编号: ");
        scanf("%s", id);
        getchar();
        bor_node = query_bor_node(bor_list, id);
        print_bor_node(bor_node);
        modify_bor_node(bor_list, bor_node);
        break;
    case '6':
        clean_bor_list(bor_list);
        break;
    case '7':
        save_bor_list(bor_list);
        break;
    case '0':
        back = 1;
        break;
    default:
        printf("输入错误, 请重新输入! \n");
    }
    if (back)
        break;
}
break;
case '3':
    fix_list = create_fix_list();
    while (1) {
        back = 0;
        print_fix();
        c = getchar();
        getchar();
        switch (c) {
            case '1':
                print_fix_list(fix_list);
                break;
            case '2':
                printf("请输入编号: ");
                scanf("%s", id);
                getchar();
                fix_node = query_fix_node(fix_list, id);
                if(fix_node != NULL)
                    print_fix_node(fix_node);

```

```

        break;
    case '3':
        fix_node = create_fix_node();
        insert_fix_node(fix_list, fix_node);
        break;
    case '4':
        printf("请输入要删除的编号: ");
        scanf("%s", id);
        getchar();
        fix_node = query_fix_node(fix_list, id);
        if(fix_node != NULL)
            delete_fix_node(fix_list, fix_node);
        break;
    case '5':
        printf("请输入编号: ");
        scanf("%s", id);
        getchar();
        fix_node = query_fix_node(fix_list, id);
        print_fix_node(fix_node);
        modify_fix_node(fix_list, fix_node);
        break;
    case '6':
        clean_fix_list(fix_list);
        break;
    case '7':
        save_fix_list(fix_list);
        break;
    case '0':
        back = 1;
        break;
    default:
        printf("输入错误, 请重新输入! \n");
    }
    if (back)
        break;
}
break;
case '4':
    mod_list = create_mod_list();
    while (1) {
        back = 0;
        print_modify();
        c = getchar();
        getchar();
        switch (c) {
            case '1':
                print_mod_list(mod_list);
                break;
            case '2':
                printf("请输入编号: ");
                scanf("%s", id);
                getchar();
                mod_node = query_mod_node(mod_list, id);
                if(mod_node != NULL)
                    print_mod_node(mod_node);
                break;
            case '3':
                mod_node = create_mod_node();

```

```

        insert_mod_node(mod_list, mod_node);
        break;
    case '4':
        printf("请输入要删除的编号: ");
        scanf("%s", id);
        getchar();
        mod_node = query_mod_node(mod_list, id);
        if(mod_node != NULL)
            delete_mod_node(mod_list, mod_node);
        break;
    case '5':
        printf("请输入编号: ");
        scanf("%s", id);
        getchar();
        mod_node = query_mod_node(mod_list, id);
        print_mod_node(mod_node);
        modify_mod_node(mod_list, mod_node);
        break;
    case '6':
        clean_mod_list(mod_list);
        break;
    case '7':
        save_mod_list(mod_list);
        break;
    case '0':
        back = 1;
        break;
    default:
        printf("输入错误, 请重新输入! \n");
    }
    if (back)
        break;
    }
    break;
case '0':
    return 0;
default:
    printf("输入错误, 请重新输入! \n");
}
}
return 0;
}

void print_main(){
    printf("\n");
    printf("+-----+\n");
    printf("| 设备信息管理系统 | \n");
    printf("+-----+\n");
    printf("1. 设备入库登记\n");
    printf("2. 设备领用与归还登记\n");
    printf("3. 设备维修登记\n");
    printf("4. 设备变更登记\n");
    printf("0. 退出系统\n");
    printf("\n");
    printf("请选择: ");
}

void print_enter(){

```



```

printf("\n");
printf("-----\n");
printf("    设备入库登记    \n");
printf("-----\n");
printf("1. 列出所有设备入库记录\n");
printf("2. 查询设备\n");
printf("3. 添加设备登记\n");
printf("4. 删除设备登记\n");
printf("5. 修改记录\n");
printf("6. 清空登记记录\n");
printf("7. 保存记录\n");
printf("0. 返回主菜单\n");
printf("\n");
printf("请选择: ");
}

void print_borrow(){
printf("\n");
printf("-----\n");
printf("    设备领用与归还登记    \n");
printf("-----\n");
printf("1. 列出所有领用/归还记录\n");
printf("2. 查询登记\n");
printf("3. 添加领用/归还登记\n");
printf("4. 删除领用/归还登记\n");
printf("5. 修改记录\n");
printf("6. 清空登记记录\n");
printf("7. 保存记录\n");
printf("0. 返回主菜单\n");
printf("\n");
printf("请选择: ");
}

void print_fix(){
printf("\n");
printf("-----\n");
printf("    设备维修登记    \n");
printf("-----\n");
printf("1. 列出所有维修记录\n");
printf("2. 查询登记\n");
printf("3. 添加维修登记\n");
printf("4. 删除维修登记\n");
printf("5. 修改记录\n");
printf("6. 清空登记记录\n");
printf("7. 保存记录\n");
printf("0. 返回主菜单\n");
printf("\n");
printf("请选择: ");
}

void print_modify(){
printf("\n");
printf("-----\n");
printf("    设备变更登记    \n");
printf("-----\n");
printf("1. 列出所有变更记录\n");
printf("2. 查询变更\n");
printf("3. 添加变更登记\n");

```

```

        printf("4. 删除变更登记\n");
        printf("5. 修改记录\n");
        printf("6. 清空登记记录\n");
        printf("7. 保存记录\n");
        printf("0. 返回主菜单\n");
        printf("\n");
        printf("请选择: ");
    }

/*entreg.h*/
#ifndef ENTREG_H
#define ENTREG_H
#define ENTER_FILE "enter.dat"
struct ent_node;
typedef struct ent_node* ent_ptr;
struct ent_node{
    char id[10];
    char type[10];
    char brand[16];
    char model[16];
    long value;
    char buy_date[10];
    ent_ptr next;
};
void clean_ent_list(ent_ptr list);
ent_ptr create_ent_list(void);
ent_ptr create_ent_node(void);
void save_ent_list(ent_ptr list);
void insert_ent_node(ent_ptr list, ent_ptr node);
void delete_ent_node(ent_ptr list, ent_ptr node);
ent_ptr query_ent_node(ent_ptr list, char* id);
void print_ent_list(ent_ptr list);
void print_ent_node(ent_ptr node);
void modify_ent_node(ent_ptr list, ent_ptr node);
#endif

/*entreg.c*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "entreg.h"
void clean_ent_list(ent_ptr list){
    ent_ptr pos, tmp;
    printf("\n");
    if (list->next == NULL) {
        printf("入库记录已经为空! \n");
        return;
    }
    pos = list->next;
    while (pos != NULL) {
        tmp = pos->next;
        free(pos);
        pos = tmp;
    }
    printf("记录已清空! \n");
}
ent_ptr create_ent_list(void){
    FILE* fp;
    ent_ptr list = NULL;

```

```

    ent_ptr node;
    list = malloc(sizeof(struct ent_node));
    list->next = NULL;
    fp = fopen(ENTER_FILE, "rb");
    if (fp == NULL)
        return list;
    while (1) {
        node = malloc(sizeof(struct ent_node));
        if (fread(node, sizeof(struct ent_node), 1, fp) == 0)
            break;
        node->next = NULL;
        insert_ent_node(list, node);
    }
    return list;
}

ent_ptr create_ent_node(void){
    ent_ptr node;

    printf("\n");
    node = malloc(sizeof(struct ent_node));
    printf("设备编号: ");
    gets(node->id);
    printf("类别: ");
    gets(node->type);
    printf("品牌: ");
    gets(node->brand);
    printf("型号: ");
    gets(node->model);
    printf("采购时间(YY/MM/DD): ");
    gets(node->buy_date);
    printf("固定资产原值: ");
    scanf("%ld", &node->value);
    getchar();
    node->next = NULL;
    return node;
}

void save_ent_list(ent_ptr list){
    FILE* fp;
    ent_ptr pos = list->next;
    fp = fopen(ENTER_FILE, "wb");
    while(pos != NULL) {
        fwrite(pos, sizeof(struct ent_node), 1, fp);
        pos = pos->next;
    }
    printf("\n");
    printf("保存成功! \n");
    fclose(fp);
}

void insert_ent_node(ent_ptr list, ent_ptr node){
    ent_ptr pos = list;
    while (pos->next != NULL)
        pos = pos->next;
    node->next = pos->next;
    pos->next = node;
}

```

```

void delete_ent_node(ent_ptr list, ent_ptr node){
    ent_ptr pos = list;
    printf("\n");
    if (pos->next == NULL) {
        printf("列表为空! \n");
        return;
    }
    while (pos->next != node && pos->next != NULL)
        pos = pos->next;
    if (pos->next != NULL) {
        pos->next = node->next;
        free(node);
        printf("删除成功! \n");
    } else
        printf("没有找到记录! \n");
}

```

```

ent_ptr query_ent_node(ent_ptr list, char* id){
    ent_ptr pos = list->next;
    while (pos != NULL && strcmp(pos->id, id) != 0)
        pos = pos->next;
    if (pos == NULL) {
        printf("\n");
        printf("没有找到记录! \n");
    }
    return pos;
}

```

```

void print_ent_list(ent_ptr list){
    ent_ptr pos = list->next;
    printf("\n");
    if (list->next == NULL) {
        printf("列表为空! \n");
        return;
    }
    printf("%-10s", "设备编号");
    printf("%-10s", "类别");
    printf("%-16s", "品牌");
    printf("%-16s", "型号");
    printf("%-10s", "采购日期");
    printf("%s ", "固定资产原值");
    printf("\n");
    while (pos != NULL) {
        printf("%-10s", pos->id);
        printf("%-10s", pos->type);
        printf("%-16s", pos->brand);
        printf("%-16s", pos->model);
        printf("%-10s", pos->buy_date);
        printf("%ld", pos->value);
        printf("\n");
        pos = pos->next;
    }
}

```

```

void print_ent_node(ent_ptr node){
    printf("\n");
    printf("设备编号: %s\n", node->id);
}

```

```

printf("类别: %s\n", node->type);
printf("品牌: %s\n", node->brand);
printf("型号: %s\n", node->model);
printf("采购日期: %s\n", node->buy_date);
printf("固定资产原值: %ld\n", node->value);
}

void modify_ent_node(ent_ptr list, ent_ptr node){
char c;
while (1) {
printf("\n");
printf("请输入要修改的属性");
printf("(1: 设备编号; 2: 类别; 3: 品牌; 4: 型号; ");
printf("5: 采购日期; 6: 固定资产原值; 0: 返回): ");
c = getchar();
getchar();/*吸收回车*/
switch (c) {
case '1':
printf("设备编号: ");
gets(node->id);
break;
case '2':
printf("类别: ");
gets(node->type);
break;
case '3':
printf("品牌: ");
gets(node->brand);
break;
case '4':
printf("型号: ");
gets(node->model);
break;
case '5':
printf("采购日期: ");
gets(node->buy_date);
break;
case '6':
printf("固定资产原值: ");
scanf("%ld", &node->value);
getchar();
break;
case '0':
return;
default:
printf("输入错误, 请重新输入! \n");
}
}
}

```

## 五、代码测试

1、当程序运行时，主菜单显示如下：

```

+-----+
| 设备信息管理系统 |
+-----+

```

1. 设备入库登记
2. 设备领用与归还登记
3. 设备维修登记
4. 设备变更登记
0. 退出系统

请选择:

- 2、此时选择相应选项会进入下一级菜单，例如选择1，回车:

请选择: 1

-----  
设备入库登记  
-----

1. 列出所有设备入库记录
2. 查询设备
3. 添加设备登记
4. 删除设备登记
5. 修改记录
6. 清空登记记录
7. 保存记录
0. 返回主菜单

请选择:

- 3、添加设备登记

请选择: 3

设备编号: 0001

类别: 笔记本

品牌: Lenovo

型号: 天逸 F41

采购时间(YY/MM/DD): 07/7/18

固定资产原值: 8600

- 4、列出所有设备入库记录

请选择: 1

设备编号	类别	品牌	型号	采购日期	固定资产原值
0001	笔记本	Lenovo	天逸 F41	07/7/18	8600
0002	路由器	TP-Link	A3180	07/7/23	130

若链表为空，则输出如下信息:

列表为空!

- 5、根据设备编号查询设备信息

请选择: 2

请输入设备编号: 0001

设备编号: 0001

类别: 笔记本

品牌: Lenovo

型号: 天逸 F41

采购日期: 07/07/18

固定资产原值: 8600

若设备不存在，需要有提示：

请选择：2

请输入设备编号：1386

没有找到记录！

#### 6、删除设备登记

请选择：4

请输入要删除的编号：0002

删除成功！

若编号不存在，则打印提示信息：

请选择：4

请输入要删除的编号：1386

没有找到记录！

#### 7、修改设备登记记录

请选择：5

请输入设备编号：0001

设备编号：0001

类别：笔记本

品牌：Lenovo

型号：天逸 F41

采购日期：07/07/18

固定资产原值：8600

请输入要修改的属性(1：设备编号；2：类别；3：品牌；4：型号；5：采购日期；6：固定

资产原值；0：返回)：

此时提示修改指定属性，例如选择 4，修改型号：

型号：昭阳 A31

输入 0 则返回菜单。

#### 8、清空列表

请选择：6

记录已清空！

#### 9、将列表保存到文件中

请选择：7

保存成功！

## 六、思考题

- 1、在编写头文件时的宏定义 `#ifndef... #define... #endif` 的作用是什么？
- 2、为什么要使用带头节点的链表而不是直接向链表的头节点中存储数据？
- 3、设备管理方面有哪些关键信息？分别使用哪些存储格式比较合理？

## 第六章 图书信息管理

本章要求设计一个图书信息管理程序，利用结构体存储每条图书信息，并使用链表存储全部图书信息；能够完成对图书信息的添加、查询、删除和对图书列表的清空等操作；可以将链表中的图书信息保存在文件中，并且可以对文件中的图书信息进行读取与显示。

### 一、实践目的

- 1、理解链表的工作原理，掌握对链表进行创建、增加、删除、查询等操作。
- 2、熟悉 C 语言中文件操作相关函数，掌握操作二进制文件的方法。
- 3、了解图书管理的相关信息及处理方法。

### 二、基本要求

- 1、使用动态链表保存图书信息，每个节点要求包含图书的编号、书名、作者、购买日期和价格信息。
- 2、可以对当前图书数据库进行增加、删除的操作，并实现按图书编号进行查询。
- 3、在每次运行时，图书数据可以从二进制文件中导入。如果对数据进行修改，可以将修改后的数据保存在二进制文件中。
- 4、系统制作完成后应实现类似图 6.1 所示界面。

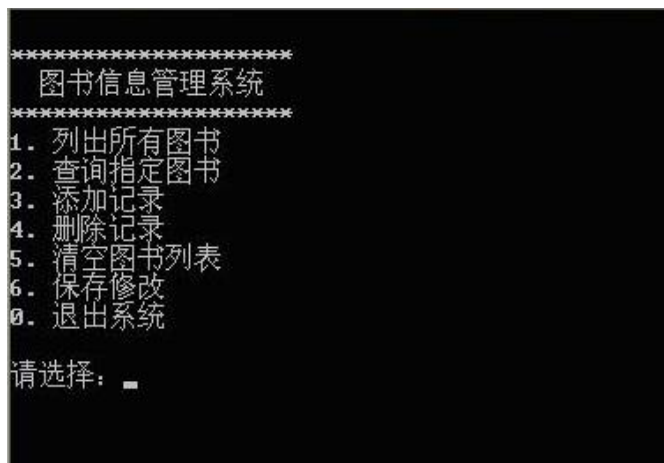


图 6.1 程序运行界面

### 三、算法分析

#### 1、数据结构

设计一个存储一条图书信息的结构体作为链表中的节点，可参考使用下述的结构体数据类型定义：

```
struct bk_node{
    char id[16];
    char name[32];
```



```

    char author[16];
    char date[16];
    float price;
    struct bk_node *next;
};

```

## 2、函数定义

```

n_ptr clean_list(n_ptr list); /*清空列表中所有内容*/
n_ptr create_list(void); /*从文件中读取列表*/
n_ptr create_node(void); /*创建一个节点*/
void save_list(n_ptr list); /*保存列表内容到文件*/
n_ptr insert_node(n_ptr list, n_ptr node); /*插入一条新纪录*/
n_ptr delete_node(n_ptr list, n_ptr node); /*删除一条记录，按节点查询*/
n_ptr query_node(n_ptr list, char* id); /*按 ID 查询，返回一条节点*/
void print_list(n_ptr list); /*打印所有条目*/
void print_node(n_ptr node); /*打印单个条目信息*/

```

## 3、处理过程

(1) 创建链表，并可以从备份文件中读取已经备份的图书信息（若文件中已存储图书信息）到链表中。在这里我们使用了不包含空头节点的链表，也就是说在头节点里存储了一条图书信息。因此在对链表进行操作时需要考虑链表头节点这个特殊状况，在编程时稍微有些麻烦。读者可自己考虑使用有空的头节点的链表来解决问题。

(2) 设计对链表相关操作的函数：包括插入一个节点，根据 ID 查找节点，删除一个节点，打印一个节点及打印整个链表，以及清空链表等。创建节点由 `n_ptr create_node(void);` 函数完成。`node` 是一个指向图书信息的节点的指针，使用 `node = malloc(sizeof(struct bk_node));` 为其分配内存空间。在从终端读入用户输入该条图书记录的所有信息。将创建完的节点插入到链表这一操作由函数 `n_ptr insert_node(n_ptr list, n_ptr node);` 完成。由于在这里我们没有使用空的头节点，因此在处理时必须增加对头节点的额外的处理：

```

n_ptr insert_node(n_ptr list, n_ptr node){
    n_ptr pos = list;
    /*当链表为 NULL 时，进行如下处理，此时链表头会发生改变*/
    if (list == NULL){
        list = node;
        return list;
    }
    while (pos->next != NULL)
        pos = pos->next;
    node->next = pos->next;
    pos->next = node;
    return list;
}

```

函数 `n_ptr query_node(n_ptr list, char* id);` 通过 ID 号来查询图书记录：

```

n_ptr query_node(n_ptr list, char* id){
    n_ptr pos = list;
    while (pos != NULL && strcmp(pos->id, id) != 0)
        pos = pos->next;.....
}

```

函数 `n_ptr delete_node(n_ptr list, n_ptr node);` 能够删除参数 `node` 所指向的节点。注意对于头节点的额外处理：

```

n_ptr delete_node(n_ptr list, n_ptr node){
    n_ptr pos = list;
    n_ptr tmp;
    if (list == NULL){

```

```

printf("列表为空! \n");
return list;
}
/*要删除的条目为表头, 表头会发生改变*/
if (pos == node){
tmp = pos->next;
free(pos);
printf("删除成功! \n");
return tmp;
}.....

```

函数 `n_ptr clean_list(n_ptr list)`; 用来清空整个链表, 注意要先判断链表是否为空:

```

n_ptr clean_list(n_ptr list){
n_ptr pos, tmp;
if (list == NULL){
printf("列表已经为空! \n");
return NULL;
}
pos = list;.....

```

### (3) 将整个链表存储的图书信息备份到文件中

在程序开始运行时读取文件信息到链表中。这里对文件的读写我们使用了 `fread` 和 `fwrite` 函数, 我们建议使用这两个函数来将结构存取于文件中。由于 IO 的效率问题, 也可以考虑使用缓冲一次读写多个结构的方法。

```

n_ptr create_list(void)
{
FILE* fp;
n_ptr list = NULL;
n_ptr node;
fp = fopen(DATA_FILE, "rb");
if (fp == NULL)
return list;.....

```

在程序退出时将整个链表保存到文件中:

```

void save_list(n_ptr list)
{
FILE* fp;
n_ptr pos = list;
fp = fopen(DATA_FILE, "wb");
while(pos != NULL)
{
fwrite(pos, sizeof(struct bk_node), 1, fp);
pos = pos->next;
}
printf("保存成功! \n");
fclose(fp);
}

```

## 四、参考代码

**/\*bklist.h\*/**

```

#ifndef BKLIST_H
#define BKLIST_H
#define DATA_FILE "data"
struct bk_node;
typedef struct bk_node* n_ptr;
struct bk_node{
char id[16];
char name[32];
char author[16];

```

```

    char date[16];
    float price;
    n_ptr next;
};
n_ptr clean_list(n_ptr list); /*清空列表中所有内容*/
n_ptr create_list(void); /*从文件中读取列表*/
n_ptr create_node(void); /*创建一个节点*/
void save_list(n_ptr list); /*保存列表内容到文件*/
n_ptr insert_node(n_ptr list, n_ptr node); /*插入一条新纪录*/
n_ptr delete_node(n_ptr list, n_ptr node); /*删除一条记录，按节点查询*/
n_ptr query_node(n_ptr list, char* id); /*按 ID 查询，返回一条节点*/
void print_list(n_ptr list); /*打印所有条目*/
void print_node(n_ptr node); /*打印单个条目信息*/
#endif
/*bklist.c*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "bklist.h"
n_ptr clean_list(n_ptr list){
    n_ptr pos, tmp;
    if (list == NULL) {
        printf("列表已经为空! \n");
        return NULL;
    }
    pos = list;
    while (pos != NULL) {
        tmp = pos->next;
        free(pos);
        pos = tmp;
    }
    printf("列表已清空! \n");
    return NULL;
}

n_ptr create_list(void){
    FILE* fp;
    n_ptr list = NULL;
    n_ptr node;
    fp = fopen(DATA_FILE, "rb");
    if (fp == NULL)
        return list;
    while (1) {
        node = malloc(sizeof(struct bk_node));
        if (fread(node, sizeof(struct bk_node), 1, fp) == 0)
            break;
        node->next = NULL;
        list = insert_node(list, node);
    }
    return list;
}

void save_list(n_ptr list){
    FILE* fp;
    n_ptr pos = list;
    fp = fopen(DATA_FILE, "wb");
    while(pos != NULL) {
        fwrite(pos, sizeof(struct bk_node), 1, fp);

```

```

        pos = pos->next;
    }
    printf("保存成功! \n");
    fclose(fp);
}

n_ptr create_node(void){
    n_ptr node;
    node = malloc(sizeof(struct bk_node));
    printf("编号: ");
    gets(node->id);
    printf("书名: ");
    gets(node->name);
    printf("作者: ");
    gets(node->author);
    printf("出版日期(YY/MM/DD): ");
    gets(node->date);
    printf("价格: ");
    scanf("%f", &node->price);
    node->next = NULL;
    getchar();
    return node;
}

n_ptr insert_node(n_ptr list, n_ptr node){
    n_ptr pos = list;
    /*当链表为 NULL 时, 进行如下处理, 此时链表头会发生改变*/
    if (list == NULL) {
        list = node;
        return list;
    }
    while (pos->next != NULL)
        pos = pos->next;
    node->next = pos->next;
    pos->next = node;
    return list;
}

n_ptr delete_node(n_ptr list, n_ptr node){
    n_ptr pos = list;
    n_ptr tmp;
    if (list == NULL) {
        printf("列表为空! \n");
        return list;
    }
    /*要删除的条目为表头, 表头会发生改变*/
    if (pos == node) {
        tmp = pos->next;
        free(pos);
        printf("删除成功! \n");
        return tmp;
    }
    while (pos->next != node && pos->next != NULL)
        pos = pos->next;
    if (pos->next != NULL) {
        pos->next = node->next;
        free(node);
        printf("删除成功! \n");
    }
}

```

```

    } else
        printf("没有找到记录! \n");
    return list;
}

n_ptr query_node(n_ptr list, char* id){
    n_ptr pos = list;
    while (pos != NULL && strcmp(pos->id, id) != 0)
        pos = pos->next;
    if (pos == NULL)
        printf("没有找到记录! \n");
    return pos;
}

void print_list(n_ptr list){
    n_ptr pos = list;
    if (list == NULL){
        printf("列表为空! \n");
        return;
    }
    printf("%-10s", "编号");
    printf("%-30s", "书名");
    printf("%-15s", "作者");
    printf("%-10s", "出版日期");
    printf("%s", "价格");
    printf("\n");
    while (pos != NULL) {
        printf("%-10s", pos->id);
        printf("%-30s", pos->name);
        printf("%-15s", pos->author);
        printf("%-10s", pos->date);
        printf("%.2f", pos->price);
        printf("\n");
        pos = pos->next;
    }
}

void print_node(n_ptr node){
    printf("编号: %s\n", node->id);
    printf("书名: %s\n", node->name);
    printf("作者: %s\n", node->author);
    printf("出版日期: %s\n", node->date);
    printf("价格: %.2f\n", node->price);
}

/*bkman.c*/
#include <stdio.h>
#include <stdlib.h>
#include "bklist.h"
void print_menu(void);

int main(int argc, char** argv){
    char c;
    char id[16];
    n_ptr list;
    n_ptr node;
    list = create_list();
    whilea (1) {
        print_menu();

```

```

c = getchar();
getchar();/*吸收回车*/
switch (c) {
    case '1':
        print_list(list);
        break;
    case '2':
        printf("请输入要查询的图书编号: ");
        scanf("%s", id);
        getchar();
        node = query_node(list, id);
        if(node != NULL)
            print_node(node);
        break;
    case '3':
        node = create_node();
        list = insert_node(list, node);
        break;
    case '4':
        printf("请输入要删除的图书编号: ");
        scanf("%s", id);
        getchar();
        node = query_node(list, id);
        if(node != NULL)
            list = delete_node(list, node);
        break;
    case '5':
        list = clean_list(list);
        break;
    case '6':
        save_list(list);
        break;
    case '0':
        return 0;
    default:
        printf("输入错误, 请重新输入! \n");
}
}
return 0;
}

void print_menu(void){
    printf("\n");
    printf("*****\n");
    printf("  图书信息管理系统\n");
    printf("*****\n");
    printf("1. 列出所有图书\n");
    printf("2. 查询指定图书\n");
    printf("3. 添加记录\n");
    printf("4. 删除记录\n");
    printf("5. 清空图书列表\n");
    printf("6. 保存修改\n");
    printf("0. 退出系统\n");
    printf("\n");
    printf("请选择: ");
}

```

## 五、代码测试

1、当程序运行时，创建链表，并从名为 data 的文件中读取已经备份的图书信息（假设文件中已存储图书信息）到链表中，显示主菜单，以下每次读取用户操作前均会打印该主菜单

\*\*\*\*\*

图书信息管理系统

\*\*\*\*\*

1. 列出所有图书
2. 查询指定图书
3. 添加记录
4. 删除记录
5. 清空图书列表
6. 保存修改
0. 退出系统

2、添加一条图书信息到链表中

请选择：3

编号：001

书名：C 程序设计

作者：Balagurusamy

出版日期(YY/MM/DD)：06/06/01

价格：35.00

3、列出链表中所有图书信息，此时已存储两条图书信息

请选择：1

编号	书名	作者	出版日期	价格
001	C 程序设计	Balagurusamy	06/06/01	35.00
002	操作系统概念	Silberschatz	04/01/01	55.00

4、根据图书编号查询图书信息

请选择：2

请输入要查询的图书编号：002

编号：002

书名：操作系统概念

作者：Silberschatz

出版日期：04/01/01

价格：55.00

若图书不存在，需要有提示：

请选择：2

请输入要查询的图书编号：003

没有找到记录！

5、从链表中删除图书信息

请选择：4

请输入要删除的图书编号：002

删除成功！

若编号不存在，则打印提示信息：

请选择：4

请输入要删除的图书编号：004

没有找到记录！

6、将链表中的图书信息存储于文件中

请选择：6

保存成功！

7、清空整个链表

请选择：5

列表已清空！

## 六、思考题

- 1、在创建链表时为何要使用 `malloc` 语句为链表申请内存空间？
- 2、在对文件操作完成后，为什么要用 `fclose()` 函数来关闭已打开的文件？
- 3、本实践是否可以使用文本文件代替二进制文件？哪一种更有优势？



## 第七章 医院信息管理

本章要求设计一个医院信息管理程序，利用结构体存储每个病人的信息和每种药品的信息，并使用链表存储全部病人信息；能够完成对医院内所有病人信息的注册、查询、删除和修改等操作，同时又能对药房内库存的药品进行查询；可以将链表中的病人信息保存在文件中，并且可以对文件中的病人信息进行读取与显示。

### 一、实践目的

- 1、掌握链表的操作，包括链表节点的创建、释放还有链表的遍历。
- 2、掌握对二进制文件的创建、增添等基本操作。
- 3、进一步熟悉 C 语言函数的使用方法，学会模块化处理问题以及多个源文件处理的方式。

### 二、基本要求

- 1、使用结构体来存储病人的信息，结构体中包括病人的 id 号码、姓名、病历以及消费信息，并用链表将所有病人信息整合。
- 2、用文件来存储链表的信息以便下次再使用该程序时能够载入病人信息。
- 3、能够实现新病人信息的注册、病人信息的查询、病人消费统计、保存链表信息、载入链表信息、查询库存等几项功能。
- 4、要求用 4 个源文件即 main.c、link.c、find.c、save\_load.c 来分别实现四大基本功能。
- 5、系统制作完成后应实现类似图 7.1 所示界面。

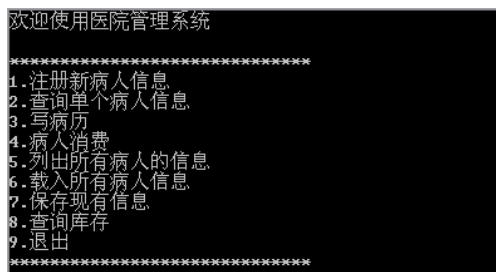


图 7.1 程序运行界面

### 三、算法分析

#### 1、数据结构

设计链表中的一个节点存储一个病人的信息，可参考使用下面的结构体类型定义：

```
struct patient{
    char id[10];
    char name[10];
    char casehist[200];
    int cost[3];
    int transfusion;
    int surgery;
    struct patient *next;
};
```

其中 `cost` 数组用来存储病人买药的个数（总共三种药），`casehist` 存储病历内容，`transfusion` 用来存储病人输液的花费，`surgery` 存储手术的花费。`struct patient *next` 用来指向下一个病人结构体的起始地址从而形成一个链表。

此外还要用到一个结构体来存储库存中当前的药品情况。

```
struct storage{
    int amount[3];
    int price[3];
};
```

2、函数定义

程序代码由 `main.c`、`link.c`、`find.c`、`save_load.c` 四个源文件组成，使用头文件 `head.h` 把四个源文件连接起来，四个文件中所包含的函数说明如表 7.1 所示。

表 7.1 文件及函数说明

源文件	函数原型	功能说明
main.c	void test()	检测有无存储病人的二进制文件
	void init(struct storage *temp)	初始化库存中的药品，参数为指向 struct storage 类型的指针
	void printmenu()	打印提示菜单
	void liststock(struct storage temp)	列出库存中的药品
	struct patient *enroll()	注册新病人信息
	void freeall(struct patient *temp)	清除链表内容，参数为指向 struct patient 类型的指针
find.c	void search(struct patient *temp)	查询单个病人的信息，参数为指向 struct patient 类型的指针
	void listall(struct patient *temp)	列出所有病人的信息，参数为指向 struct patient 类型的指针
link.c	struct patient *insert(struct patient *head,struct patient *rear)	加入新节点，参数为指向 struct patient 类型的指针
	void modify(struct patient *temp)	修改病人病历，参数为指向 struct patient 类型的指针
	void buy(struct patient *temp,struct storage *s)	用于病人的消费处理，参数为指向 struct patient 类型的指针与指向 struct storage 的指针
save_load.c	void save(struct patient *head)	将链表信息保存到文件中，参数为指向 struct patient 类型的指针
	struct patient *load(struct patient *head)	将文件信息写入到链表中，参数为指向 struct patient 类型的指针

3、处理过程：

- （1）检查存储病人信息的文件是否存在，若不存在则创建一个新文件并设置 `flag` 值为 1。以标记现在可以对链表进行操作。
- （2）初始化库存令巴米尔、感冒灵、病毒唑 3 种药的数量分别为 20、20、10，价格分别为 5、9、16。

(3) 输出提示菜单让用户选择要操作的项目。(下面以写病历和载入链表信息为例介绍处理过程。)

1) 写病历前先检查是否有可用的链表。(此标记为 `flag` 的值, 若为 1 则代表可以修改病历, 否则要求用户重新作出选择。) 确认有可用链表之后进入 `modify` 函数。要求用户输入病人的 `id` 号码, 然后在链表中寻找该病人所对应的节点, 若无此病人的 `id` 则直接输出提示信息后返回, 找到后先输出原来的病历内容, 然后提示用户输入当前日期和新添加的病历内容, 调用 `strcmp` 函数将新内容追加到原来的 `casehist` 字符串数组中。

```
printf("原有病历: %s\n",temp->casehist);
printf("输入当前日期 年份");
gets(a);
printf("输入当前月份");
gets(b);
printf("输入当前日子");
gets(c);
strcat(temp->casehist,a);
strcat(temp->casehist,"/");
strcat(temp->casehist,b);
strcat(temp->casehist,"/");
strcat(temp->casehist,c);
strcat(temp->casehist," ");
printf("请输入新病历的内容\n");/*是否要    加入    增添病历的功能*/
gets(d);
strcat(temp->casehist,d);
.....
```

2) 载入链表信息时首先定义一个文件指针然后以只读方式打开文件若失败则输出提示信息后返回, 成功后调用 `malloc` 函数来申请一块内存, 用 `fread` 函数将文件内容写入到节点中, 随后调用 `insert` 函数将该节点加入链表, 如此循环下去直到 `fread` 函数返回 0 值, 即文件已经读到了末尾时, 再将链表的头指针返回给主函数即可。其中要注意在返回主函数指针前一定要将 `flag` 值置为 1 用来代表已有可用链表。

```
if((fp=fopen("data","r"))==0){
    printf("文件不存在!");
    return head;
}
while(1) {
    rear=(struct patient *)malloc(sizeof(struct patient));
    if(fread(rear,sizeof(struct patient),1,fp))
    {
        head=insert(head,rear);
    }
    else
    {
        free(rear);
        flag=1;
        break;
    }
}
```

#### 四、参考代码

**/\*head.h\*/**

```

#ifndef FIRST
#include<stdlib.h>
#include<stdio.h>
#include<string.h>
struct patient{
    char id[10];
    char name[10];
    char casehist[200];
    int cost[3];
    int transfusion;
    int surgery;
    struct patient *next;
};
struct storage{
    int amount[3];
    int price[3];
};
int flag;
void search(struct patient *temp);
void listall(struct patient *temp);
struct patient *insert(struct patient *head,struct patient *rear);
void modify(struct patient *temp);
void buy(struct patient *temp,struct storage *s);
struct patient *load(struct patient *head);
void save(struct patient *head);
#define FIRST
#endif
/*find.c*/
#include "head.h"
void search(struct patient *temp){
    char t[10];
    int sum;
    printf("请输入病人的 id\n");
    gets(t);
    while(temp&&strcmp(temp->id,t))    {
        temp=temp->next;
    }
    if(temp) {
        printf("id:  %s\n",temp->id);
        printf("\n 姓名: %s",temp->name);
        printf("\n 病历:  \n%s",temp->casehist);
        printf("\n 消费记录:\n");
        if(temp->cost[0]){
            printf("巴米尔      %d 个\n",temp->cost[0]);
        }
        if(temp->cost[1]){
            printf("感冒灵      %d 个\n",temp->cost[1]);
        }
        if(temp->cost[2]){
            printf("病毒唑      %d 个\n",temp->cost[2]);
        }
        if(temp->transfusion) {
            printf("输液费      %d\n",temp->transfusion);
        }
        if(temp->surgery){
            printf("手术费      %d\n",temp->surgery);
        }
        sum=temp->cost[0]*5+temp->cost[1]*9+temp->cost[2]*16+temp->transfusion+temp->

```

```

surgery;
        printf("总费用      %d 元",sum);
    }
    else {
        printf("无该病人的信息!\n");
    }
}

void listall(struct patient *temp){
    printf("id      姓名\n");
    while(temp) {
        printf("%s      %s\n",temp->id,temp->name);
        temp=temp->next;
    }
}
/*link.c*/
#include "head.h"
struct patient *insert(struct patient *head,struct patient *rear) { /*!!!! 可以出现重复 id*/
    struct patient *temp;
    if(head) {
        temp=head;
        while(temp->next){
            temp=temp->next;
        }
        temp->next=rear;
        rear->next=0;
    }
    else {
        head=rear;
        rear->next=0;
    }
    return head;
}

void modify(struct patient *temp){
    char t[10],d[200];
    char a[4],b[2],c[2];
    printf("请输入病人的 id\n");
    gets(t);
    while(temp&&strcmp(temp->id,t)){
        temp=temp->next;
    }
    if(temp){
        printf("原有病历:  %s\n",temp->casehist);
        printf("输入当前日期  年份");
        gets(a);
        printf("输入当前月份");
        gets(b);
        printf("输入当前日子");
        gets(c);
        strcat(temp->casehist,a);
        strcat(temp->casehist,"/");
        strcat(temp->casehist,b);
        strcat(temp->casehist,"/");
        strcat(temp->casehist,c);
        strcat(temp->casehist," ");
        printf("请输入新病历的内容\n");/*是否要增添病历的功能*/
        gets(d);
    }
}

```

```

        strcat(temp->casehist,d);
        strcat(temp->casehist,"\n");
    }
    else{
        printf("无此病人信息!\n");
    }
}

void buy(struct patient *temp,struct storage *s){
    char t[10];
    int a,b,c,d,e;
    int flag1;
    printf("请输入病人的 id\n");
    gets(t);
    while(temp&&strcmp(temp->id,t)){
        temp=temp->next;
    }
    if(temp){
        flag1=0;
        do{
            printf("请输入买入巴米尔的个数:\n");
            scanf("%d",&a);
            flag1=0;
            if(a>s->amount[0]){
                printf("库存不足!\n");
                flag1=1;
            }
        }while(flag1);
        flag1=0;
        do{
            printf("请输入买入感冒灵的个数:\n");
            scanf("%d",&b);
            flag1=0;
            if(b>s->amount[1]){
                printf("库存不足!\n");
                flag1=1;
            }
        }while(flag1);
        flag1=0;
        do{
            printf("请输入买入病毒唑的个数:\n");
            scanf("%d",&c);
            flag1=0;
            if(c>s->amount[2]){
                printf("库存不足!\n");
                flag1=1;
            }
        }while(flag1);
        temp->cost[0]+=a;
        temp->cost[1]+=b;
        temp->cost[2]+=c;
        printf("请输入输液费\n");
        scanf("%d",&d);
        temp->transfusion+=d;
        printf("请输入手术费\n");
        scanf("%d",&e);
        temp->surgery+=e;
        s->amount[0]-=a;

```

```

        s->amount[1]-=b;
        s->amount[2]-=c;
        printf("\n 购买成功!\n");
    }
    else{
        printf("无此病人信息!\n");
    }
}
/*main.c*/
#include "head.h"
void test();
void init(struct storage *temp);
void printmenu();
void liststock(struct storage temp);
struct patient *enroll();
void freeall(struct patient *temp);
void main(){
    struct patient *head=0,*rear;
    struct storage stock;
    char ch;
    printf("欢迎使用医院管理系统\n");
    test();
    init(&stock);
    while(1) {
        printmenu();
        fflush(stdin);
        ch=getchar();
        fflush(stdin);
        switch(ch){
            case '1':
                if(flag){
                    rear=enroll();
                    head=insert(head,rear);
                    printf("注册成功!\n");
                }
                else {
                    printf("您没有载入信息不能注册!\n");
                }
                break;
            case '2':
                if(flag){
                    search(head);
                }
                else {
                    printf("您没有载入信息不能进行查询!\n");
                }
                break;
            case '3':
                if(flag){
                    modify(head);
                }
                else {
                    printf("您没有载入信息不能修改病历!\n");
                }
                break;
            case '4':
                if(flag){
                    buy(head,&stock);

```

```

        }
        else{
            printf("您没有载入信息不能进行消费!\n");
        }
        break;
    case '5':
        if(flag){
            listall(head);
        }
        else{
            printf("您没有载入信息!\n");
        }
        break;
    case '6':
        freeall(head);
        head=0;
        head=load(head);
        break;
    case '7':
        if(flag){
            save(head);
        }
        else {
            printf("您没有载入信息!\n");
        }
        break;
    case '8':
        liststock(stock);
        break;
    case '9':
        printf("谢谢使用!\n");
        exit(0);
        break;
    default:
        printf("输入有误!\n");
        break;
    }
}
}

```

```

void printmenu(){
    printf("\n");
    printf("*****\n");
    printf("1.注册新病人信息\n");
    printf("2.查询单个病人信息\n");
    printf("3.写病历\n");
    printf("4.病人消费\n");
    printf("5.列出所有病人的信息\n");
    printf("6.载入所有病人信息\n");
    printf("7.保存现有信息\n");
    printf("8.查询库存\n");
    printf("9.退出\n");
    printf("*****\n");
}

```

```

void test(){
    FILE *fp;
    if((fp=fopen("data","r"))==0){

```



```

        printf("这是您第一次使用该系统在退出时不要忘了保存信息\n");
        fp=fopen("data","w");
        flag=1;
        fclose(fp);
    }
}

void init(struct storage *temp){
    temp->amount[0]=20;
    temp->amount[1]=20;
    temp->amount[2]=10;
    temp->price[0]=5;
    temp->price[1]=9;
    temp->price[2]=16;
}

void liststock(struct storage temp){
    int i;
    printf("药品名      数量      单价\n");
    for(i=0;i<3;i++){
        switch(i) {
            case 0:
                printf("巴米尔      %d      %d\n",temp.amount[i],temp.price[i]);
                break;
            case 1:
                printf("感冒灵      %d      %d\n",temp.amount[i],temp.price[i]);
                break;
            case 2:
                printf("病毒唑      %d      %d\n",temp.amount[i],temp.price[i]);
                break;
        }
    }
}

struct patient *enroll(){
    struct patient *rear;
    rear=(struct patient *)malloc(sizeof(struct patient));
    printf("请输入病人 id:\n");
    scanf("%s",rear->id);
    printf("请输入病人姓名\n");
    scanf("%s",rear->name);
    strcpy(rear->casehist,"");
    rear->cost[0]=0;
    rear->cost[1]=0;
    rear->cost[2]=0;
    rear->surgery=0;
    rear->transfusion=0;
    return rear;
}

void freeall(struct patient *temp){
    struct patient *t;
    while(temp) {
        t=temp;
        temp=temp->next;
        free(t);
    }
}

```

```

/*save_load.c*/
#include "head.h"
struct patient *load(struct patient *head){
    FILE *fp;
    struct patient *rear;
    if((fp=fopen("data","r"))==0){
        printf("文件不存在!");
        return head;
    }
    while(1) {
        rear=(struct patient *)malloc(sizeof(struct patient));
        if(fread(rear,sizeof(struct patient),1,fp)){
            head=insert(head,rear);
        }
        else {
            free(rear);
            flag=1;
            break;
        }
    }
    fclose(fp);
    return head;
}

void save(struct patient *head){
    FILE *fp;
    struct patient *temp;
    temp=head;
    if((fp=fopen("data","w"))==0) { /*need to be check*/
        printf("文件已损坏!\n");
    }
    else {
        while(temp) {
            fwrite(temp,sizeof(struct patient),1,fp);
            temp=temp->next;
        }
    }
    fclose(fp);
}

```

## 五、测试代码

1、假设这是第一次使用该程序则显示如下：

欢迎使用医院管理系统

\*\*\*\*\*

- 1.注册新病人信息
- 2.查询单个病人信息
- 3.写病历
- 4.病人消费
- 5.列出所有病人的信息
- 6.载入所有病人信息
- 7.保存现有信息
- 8.查询库存
- 9.退出

\*\*\*\*\*

#### (1) 输入 1

请输入病人 id:1

请输入病人姓名:张三

注册成功!

#### (2) 输入 4

请输入病人 id:1

请输入买入巴米尔的个数:2

请输入买入感冒灵的个数:1

请输入买入病毒唑的个数:0

请输入输液费:10

请输入手术费:0

购买成功!

#### (3) 输入 3

请输入病人 id:1

原有病历:

输入当前日期 年份:2008

输入当前月份:1

输入当前日子:1

请输入新病历的内容:感冒

#### (4) 输入 2

请输入病人 id:1

id: 1

姓名:张三

病历:

2008/1/1: 感冒

消费记录:

巴米尔 2 个

感冒灵 1 个

输液费 10 元

总费用 29 元

## 六、思考题

- 1、该程序为什么要用模块化设计，如果不这样设计会有什么弊端？
- 2、是否可以出现多个重复的 id 被注册的情况，若有应怎么处理？
- 3、程序中运用了一个名为 flag 的全局变量，可否不用全局变量就能实现参数 flag 的功能？

## 第八章 超市信息管理

本章要求设计一个超市管理程序，利用结构体存储货物信息和所购买商品的信息，并使用链表保存购物车信息；能够完成建立库存信息，对购物车添加商品，结算并修改库存等操作；可以将库存信息保存在文件中，并且可以对文件中的库存信息进行读取与显示。

### 一、实践目的

- 1、掌握结构体的定义和使用。
- 2、理解链表的工作原理，掌握对链表的基本操作。
- 3、熟悉 C 语言中文件操作相关函数，掌握操作二进制文件的方法。
- 4、了解超市管理的相关信息及处理方法。

### 二、基本要求

- 1、使用两个不同的结构体分别保存货物信息和购物车中的商品信息，可在商品信息结构体中嵌套货物信息结构体。
- 2、将货物信息写入文件保存，在每次运行时，货物信息可以从二进制文件中导入，并在结算后更新。
- 3、建立一个链表保存购物车信息，可以实现商品的添加和显示当前购物列表。
- 4、系统制作完成后应实现类似图 8.1 所示界面。

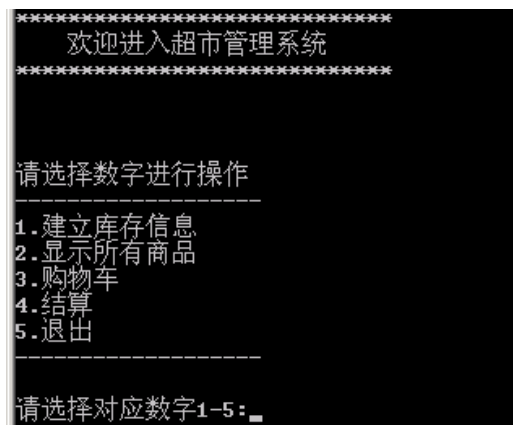


图 8.1 程序运行界面

### 三、算法分析

#### 1、数据结构

(1) 定义一个结构体存储货物信息，可参考使用下述的结构体数据类型定义：

```
struct item{
    char brand[20];
    char id[10];
    float in_price;
    float out_price;
```

```
int storage;
};
```

(2) 设计一个结构体存储购物车中的一条商品信息，该结构体嵌套了货物信息的结构体 item，可参考下面的定义方式：

```
struct item_node{
    struct item wanted;
    int amount;
    struct item_node *next;
};
```

2、函数定义

程序由头文件 market.h 和源文件 market.c、establish.c、shoppingcart.c、calculate.c，四个源文件中具体包含的函数如表 8.1 所示。

表 8.1 文件及函数说明

源文件	函数原型	功能说明
market.c	int menu()	打印主菜单，并输入选择
	void dis_all()	显示库存信息
establish.c	void establish()	建立库存信息
shoppingcart.c	void shop_cart()	对购物车进行操作
	int cart_menu()	打印购物车菜单，并返回选择
	void display()	显示购物列表
	void add()	添加商品至购物车
calculate.c	void calculate()	计算所购商品价格并修改库存

3、处理过程

(1) 创建库存信息：设计函数 void establish(); 逐个输入货物信息至结构数组 goods[NUM]，再将数组写入文件。

(2) 对购物车进行相关操作：定义指针 struct item\_node \*cart 来保存链表，若用户选择向购物车添加商品，则新建一个节点，将该商品信息存入节点中，再追加到链表的末尾。显示购物信息时，若指针不为空，逐一显示节点内容。

(3) 结算：先调用 display()显示购物清单，逐一读取链表 cart 中每一个节点的数据，计算价格，并修改结构体数组 goods。打印显示总数，并输入实付金额，计算找零。最后将修改后的结构体数组 goods 写入文件。

四、参考代码

```
//Market.h
#ifndef MARKET_H
#define MARKET_H
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define NUM 10
struct item{
```

```

        char brand[20];
        char id[10];
        float in_price;
        float out_price;
        int storage;
    };
    struct item_node{
        struct item wanted;
        int amount;
        struct item_node *next;
    };
    int menu();
    void establish();
    void dis_all();
    void shop_cart();
    int cart_menu();
    void add();
    void calculate();
    void display();
    struct item goods[NUM];
    struct item_node *cart;
#endif

//Market.c
#include "Market.h"
main(){
    printf("*****\n");
    printf("    欢迎进入超市管理系统\n");
    printf("*****\n\n");
    while(1){
        switch(menu()){
            case 1:
                establish();
                break;
            case 2:
                dis_all();
                break;
            case 3:
                shop_cart();
                break;
            case 4:
                calculate();
                break;
            case 5:
                printf("感谢使用， 再见!\n");
                exit(0);
        }
    }
}

int menu(){
    char str[5];
    int select;
    printf("\n\n 请选择数字进行操作\n");
    printf("-----\n");
    printf("1.建立库存信息\n");
    printf("2.显示所有商品\n");
    printf("3.购物车\n");

```

```

printf("4.结算\n");
printf("5.退出\n");
printf("-----\n\n");
printf("请选择对应数字 1-5:");
while(1){
    fflush(stdin);
    gets(str);
    select=atoi(str);
    if(select<1||select>5)
        printf("输入错误, 请重新输入: ");
    else
        break;
}
return select;
}

void dis_all(){
    int i;
    FILE *fp;
    fp=fopen("goods","r");
    for(i=0;(fread(goods+i,sizeof(struct item),1,fp))!=0;i++){
        printf("-----\n");
        printf("货号      品名      单价      库存量\n");
        printf("%10s%20s%7.2f%8d\n",goods[i].id,goods[i].brand,goods[i].out_price,goods[i].storage);
    }
    fclose(fp);
}

//establish.c
//建立库存信息文件
#include "Market.h"
void establish(){
    FILE *fp;
    int i;
    printf("请依次输入货物信息:\n");
    printf("-----\n");
    for(i=0;i<NUM;i++){
        printf("品名: ");
        fflush(stdin);
        gets(goods[i].brand);
        printf("货号: ");
        fflush(stdin);
        gets(goods[i].id);
        printf("进价: ");
        fflush(stdin);
        scanf("%f",&goods[i].in_price);
        printf("售价: ");
        fflush(stdin);
        scanf("%f",&goods[i].out_price);
        printf("数量: ");
        fflush(stdin);
        scanf("%d",&goods[i].storage);
        printf("\n");
    }
    if((fp=fopen("goods","w"))==NULL){
        printf("创建文件失败。 \n");
        return;
    }
    fwrite(goods,sizeof(struct item),NUM,fp);
}

```

```

        fclose(fp);
    }

//shoppingcart.c
//购物车
#include "Market.h"
void shop_cart(){
    while(1){
        switch(cart_menu()){
            case 1:
                display();
                break;
            case 2:
                add();
                break;
            case 3:
                return;
        }
    }
}

int cart_menu(){
    char str[5];
    int select;
    printf("\n 请选择操作\n");
    printf("-----\n");
    printf("1.显示当前购物列表\n");
    printf("2.添加商品\n");
    printf("3.退出\n");
    printf("-----\n\n");
    printf("请选择对应数字 1-4\n");
    while(1){
        fflush(stdin);
        gets(str);
        select=atoi(str);
        if(select<1||select>3)
            printf("输入错误，请重新输入： ");
        else
            break;
    }
    return select;
}

void display(){
    struct item_node *p=cart;
    if(p==NULL){
        printf("购物车为空\n");
        return;
    }
    while(p!=NULL){
        printf("-----\n");
        printf("货号      品名      单价      数量\n");
        printf("%10s%20s%7.2f%8d\n",p->wanted.id,p->wanted.brand,p->wanted.out_price,p->amount);
        p=p->next;
    }
}

```



```

void add(){
    FILE *fp;
    int i,n;
    char str[20];
    char choice1,choice2;
    struct item_node *p,*p1;
    do{
        printf("输入所需物品的名称或货号: \n");
        fflush(stdin);
        gets(str);
        if((fp=fopen("goods","r"))==NULL){
            printf("打开文件失败\n");
            continue;
        }
        for(i=0;fread(goods+i,sizeof(struct item),1,fp)!=0;i++){
            if((strcmp(goods[i].brand,str)==0||strcmp(goods[i].id,str)==0)&&goods[i].storage!=0){
                printf("已找到所需物品: \n");
                printf("-----\n");
                printf("货号      品名      单价      库存量\n");
                printf("%10s%20s%7.2f%8d\n",goods[i].id,goods[i].brand,goods[i].out_price,goods[i].storage);
                printf("请输入所需数量: ");
                scanf("%d",&n);
                if(n>goods[i].storage){
                    printf("库存不足\n");
                    break;
                }
                printf("\n 是否购买? (Y/N)");
                fflush(stdin);
                choice1=getchar();
                if(choice1=='Y'||choice1=='y'){
                    p1=(struct item_node*)malloc(sizeof(struct item_node));
                    if(p1==NULL){
                        printf("内存申请失败!\n");
                        exit(1);
                    }
                    p1->amount=n;
                    p1->wanted=goods[i];
                    p1->next=NULL;
                    p=cart;
                    if(cart==NULL)
                        cart=p1;
                    else {
                        while(p->next!=NULL)
                            p=p->next;
                        p1->next=p->next;
                        p->next=p1;
                    }
                }
            }
            break;
        }
    }
    if(i==NUM)
        printf("未找到所需物品\n");
    fclose(fp);
    printf("是否继续购物? (Y/N)");
    fflush(stdin);
    choice2=getchar();
}while(choice2=='Y'||choice2=='y');

```

```

}
//calculate.c
//结算
#include "Market.h"
void calculate(){
    float total=0,pay;
    struct item_node *p;
    int i;
    FILE *fp;
    printf("以下是购物清单： \n");
    display();
    if((fp=fopen("goods","r"))==NULL){
        printf("打开文件失败。 \n");
        return;
    }
    for(i=0;(fread(goods+i,sizeof(struct item),1,fp))!=0;i++)
        ;
    fclose(fp);

    p=cart;
    while(p!=NULL){
        total+=p->wanted.out_price*p->amount;
        for(i=0;strcmp(goods[i].id,p->wanted.id)!=0;i++)
            ;
        goods[i].storage-=p->amount;
        p=p->next;
    }
    printf("总计   %7.2f",total);
    printf("\n 输入实付金额： ");
    scanf("%f",&pay);
    printf("实付： %7.2f   找零：   %7.2f",pay,pay-total);
    if((fp=fopen("goods","w"))==NULL){
        printf("打开文件失败。 \n");
        return;
    }
    fwrite(goods,sizeof(struct item),NUM,fp);
    fclose(fp);
}

```

## 五、代码测试

1、当程序运行时，显示主菜单，以下每次读取用户操作前均会打印该主菜单：

```
*****
```

欢迎进入超市管理系统

```
*****
```

请选择数字进行操作

```
-----
```

- 1.建立库存信息
- 2.显示所有商品
- 3.购物车
- 4.结算
- 5.退出

```
-----
```

请选择对应数字 1-5:

2、建立库存信息

请依次输入货物信息:

品名：雕牌洗衣粉

货号：ry\_007001

进价：2.5

售价：4.5

数量：250

### 3、显示库存列表

请选择对应数字 1-5:2

库存商品

货号	品名	单价	库存量
ry_007001	雕牌洗衣粉	4.50	250

货号	品名	单价	库存量
ry_007002	佳洁士牙刷	2.50	160

货号	品名	单价	库存量
sp_008001	伊利纯牛奶	1.40	100

货号	品名	单价	库存量
sp_008002	可比克薯片	4.50	310

货号	品名	单价	库存量
wj_009001	白雪修正液	2.50	350

货号	品名	单价	库存量
hz_010001	飘柔日常护理	9.90	90

货号	品名	单价	库存量
hz_010002	李医生面膜	14.50	65

货号	品名	单价	库存量
sp_007003	红富士苹果	2.50	100

货号	品名	单价	库存量
sp_007004	齐云山枣糕	6.60	255

货号	品名	单价	库存量
wj_009002	真彩笔替芯	0.90	100

### 4、进入购物车

请选择对应数字 1-5:3

请选择操作

1.显示当前购物列表

2.添加商品

3.退出

请选择对应数字 1-4

### 5、向链表中添加商品

请选择对应数字 1-4: 2

输入所需物品的名称或货号:

飘柔日常护理

已找到所需物品:

货号	品名	单价	库存量
hz_010001	飘柔日常护理	9.90	90

请输入所需数量: 2

是否购买? (Y/N)y

是否继续购物? (Y/N)n

若商品不存在, 则打印提示信息:

输入所需物品的名称或货号:

清风面巾纸

未找到所需物品

是否继续购物? (Y/N)

#### 6、显示当前购物列表

请选择对应数字 1-4: 1

货号	品名	单价	数量
hz_010001	飘柔日常护理	9.90	2
sp_007004	齐云山枣糕	6.60	3

#### 7、结算

请选择对应数字 1-5:4

以下是购物清单:

货号	品名	单价	数量
hz_010001	飘柔日常护理	9.90	2
sp_007004	齐云山枣糕	6.60	3

总计 39.60

输入实付金额: 50

实付: 50.00 找零: 10.40

## 六、思考题

- 1、对文件有哪些基本操作?
- 2、如何完善购物车功能, 实现物品的删除呢? (提示: 即对链表进行节点删除)
- 3、建立一个新文件, 用以记录商品的销售状况。

## 第九章 票务信息管理

本章要求设计一个票务信息管理程序，要求实现查询、购买、退订票务信息等功能；使用相同的结构体来储存票务和车次信息；使用指针来实现各种车票信息的添加、删除、查询、修改等操作。

### 一、实践目的

- 1、掌握结构体数组的定义及使用方法。
- 2、会使用指针通过函数对结构体进行修改、删除等工作。
- 3、加深理解模块化的编程思想，将不同功能通过子函数实现
- 4、了解票务管理过程中所需要处理的信息以及相关的处理方法。

### 二、基本要求

- 1、本章要求设计一个票务信息管理程序，要求实现 1,查询车票 2,购买车票 3,退订车票 4,退出程序，从而完成程序所需功能。
- 2、要求使用结构体来存储票务信息。
- 3、使用指针来实现对车票信息的显示、查询、增加、删除等操作。
- 4、系统制作完成后应实现类似图 9.1 所示界面。

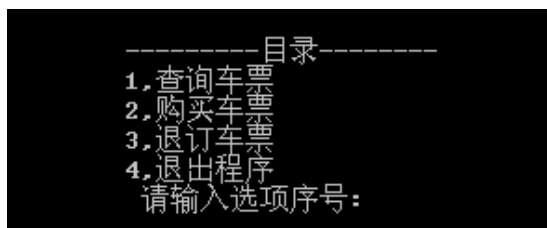


图 9.1 程序运行界面

### 三、算法分析

#### 1、数据结构

程序需要设计 2 个结构体来满足存储车票信息的需要，结构体设计可参考下例：

```
typedef struct traininfo{ //车次结构体
    char station[10][10]; //记录车站信息
    char tnumber[20]; //车次号
    float price; //车票价格
    int lnumber; //生于票量
    char time[20]; //到站时间
}pemp;
struct ticket{ //车票信息结构体
    char buyer[10]; //购票人姓名
    char source[30]; //起始站
    char dest[30]; //终点站
    int bnumber; //购买的张数
};
```

## 2、函数定义

程序源文件中具体包含的函数如表 9.1 所示。

### 表 9.1 文件及函数说明

源文件	函数原型	功能说明
ticket.c	void menu(pemp *pointtr,struct ticket *pointtk)	打印主菜单
	void buy(pemp *pointtr,struct ticket *pointtk)	购买车票
	void returtn(pemp *pointtr,struct ticket *pointtk)	退票
	void check(pemp *pointtr,struct ticket *pointtk)	查看某起始地目的地车票信息
	void show(struct ticket *pointtk,pemp *pointtr)	显示车票信息
	void exit()	退出程序

### 3、处理过程

[illegible]

```

scanf("%s",choice);
if((strcmp(choice,"1")==0){
    printf("请输入您要购买的张数:\n");{//输入张数
        scanf("%d",&buynumber);
    }
    if(((pointtr->lnumber)-buynumber)>=0){
        printf("购买成功! 返回菜单\n");
        pointtk->bnumber=buynumber;
        (pointtr->lnumber)-=buynumber;
        menu(pointtr,pointtk);
    }
    else{//若车票不够, 给出提示信息
        printf("车票已售完或不足, 不能购买! 返回菜单\n");
        menu(pointtr,pointtk);
    }
}
else if((strcmp(choice,"2")==0){//放弃购买
    printf("放弃购买, 返回菜单\n");
    menu(pointtr,pointtk);
}
else
{ menu(pointtr,pointtk);
  printf("错误,返回菜单\n");
  menu(pointtr,pointtk);
}
}
}

```

(3) 退订票务程序 void returnt(pemp \*pointtr, struct ticket \*pointtk){

```

char choice[10];//记录选项
printf("您确定要退票吗?1.是 2.否\n");//确认是否退票
scanf("%s",choice);
if((strcmp(choice,"1")==0){
    printf("退票成功! 返回菜单\n");
    pointtr->lnumber+=pointtk->bnumber;//票数增加
    menu(pointtr,pointtk);//回到菜单
}
else if((strcmp(choice,"2")==0){
    printf("放弃退票, 返回菜单\n");//放弃退票
    menu(pointtr,pointtk);
}
else{
    menu(pointtr,pointtk);
    printf("错误,返回菜单\n");
    menu(pointtr,pointtk);
}
}

```

(4) 显示信息 void show(struct ticket \*pointtk,pemp \*pointtr){

```

printf("车次:      ");//依次输出信息
printf("T11");
printf("起始地----->目的地:      ");
printf("%s----->",pointtk->source);
printf("%s\n",pointtk->dest);
printf("到达时间:      ",pointtk->source);
printf("%s\n",pointtr->time);
printf("票价:      ");
printf("%10.2f\n",pointtr->price);
printf("车票数量:      ");

```

```

        printf("%d\n",pointtr->lnumber);
    }
(5) 按起始地目的地查找 void check(pemp *pointtr,struct ticket *pointtk){
    int i,j,count=0;
    char sour[10];
    char dest[10];
    printf("请输入起始地: \n");//输入起始地
    scanf("%s",sour);
    printf("请输入目的地: \n");//输入目的地
    scanf("%s",dest);
    for(i=0;i<NUM;i++){
        if(strcmp(pointtr->station[i],sour)==0){//查找第一个车站
            for(j=0;j<NUM;j++){
                if(strcmp(pointtr->station[j],dest)==0){//查找第二个车站
                    strcpy(pointtk->source,pointtr->station[i]);//将车站写入车票结构体
                    strcpy(pointtk->dest,pointtr->station[j]);
                    show(pointtk,pointtr);
                    count++;
                    break;
                }
            }
        }
    }
    if(count==0){
        printf("无信息! \n");
        strcpy(pointtk->source,"0");//若没有找到, 车站为空
        strcpy(pointtk->dest,"0");
        return;
    }
}
}

```

#### 四、参考代码

```

/*ticket.c*/
#include<stdio.h>
#include<string.h>
#define NUM 100
typedef struct traininfor{
    char station[10][10];
    char tnumber[20];
    float price;
    int lnumber;
    char time[20];
}pemp;
struct ticket{
    char buyer[10];
    char source[30];
    char dest[30];
    int bnumber;
};
void menu(pemp *pointtr,struct ticket *pointtk);/*菜单显示程序*/
void buy(pemp *pointtr,struct ticket *pointtk);//
void returnt(pemp *pointtr,struct ticket *pointtk);/**/
void check(pemp *pointtr,struct ticket *pointtk);
void show(struct ticket *pointtk,pemp *pointtr);
void exit();/*出程序*/

```



[illegible]

```

void check(pemp *pointtr,struct ticket *pointtk){
    int i,j,count=0;
    char sour[10];
    char dest[10];
    printf("请输入起始地: \n");
    scanf("%s",sour);
    printf("请输入目的地: \n");
    scanf("%s",dest);
    for(i=0;i<NUM;i++){
        if(strcmp(pointtr->station[i],sour)==0){
            for(j=0;j<NUM;j++){
                if(strcmp(pointtr->station[j],dest)==0){
                    strcpy(pointtk->source,pointtr->station[i]);
                    strcpy(pointtk->dest,pointtr->station[j]);
                    show(pointtk,pointtr);
                    count++;
                    break;
                }
            }
        }
    }
    if(count==0){
        printf("无信息! \n");
        strcpy(pointtk->source,"0");
        strcpy(pointtk->dest,"0");
        return;
    }
}

void buy(pemp *pointtr,struct ticket *pointtk){
    char choice[10];
    int buynumber;
    printf("请输入您的姓名:\n");
    scanf("%s",pointtk->buyer);
    check(pointtr,pointtk);
    if((strcmp(pointtk->dest,"0")==0){
        printf("无此站! 返回菜单! \n");
        menu(pointtr,pointtk);
    }
    else{
        printf("您确定购买此车票吗? 1.是 2.否\n");
        scanf("%s",choice);
        if((strcmp(choice,"1")==0){
            printf("请输入您要购买的张数:\n");{
                scanf("%d",&buynumber);
            }
            if(((pointtr->lnumber)-buynumber)>=0){
                printf("购买成功! 返回菜单\n");
                pointtk->bnumber=buynumber;
                (pointtr->lnumber)-=buynumber;
                menu(pointtr,pointtk);
            }
            else{
                printf("车票已售完或不足, 不能购买! 返回菜单\n");
                menu(pointtr,pointtk);
            }
        }
        else if((strcmp(choice,"2")==0){

```

```

        printf("放弃购买，返回菜单\n");
        menu(pointtr,pointtk);
    }
    else
    { menu(pointtr,pointtk);
      printf("错误,返回菜单\n");
      menu(pointtr,pointtk);
    }
}
}

void returnt(pemp *pointtr, struct ticket *pointtk){
    char choice[10];
    printf("您确定要退票吗?1.是 2.否\n");
    scanf("%s",choice);
    if((strcmp(choice,"1")==0){
        printf("退票成功! 返回菜单\n");
        pointtr->lnumber+=pointtk->bnumber;
        menu(pointtr,pointtk);
    }
    else if((strcmp(choice,"2")==0){
        printf("放弃退票，返回菜单\n");
        menu(pointtr,pointtk);
    }
    else{
        menu(pointtr,pointtk);
        printf("错误,返回菜单\n");
        menu(pointtr,pointtk);
    }
}
}

```

## 五、代码测试

1、当程序运行时，主菜单显示如下：

-----目录-----

1,查询车票

2,购买车票

3,退订车票

4,退出程序

请输入选项序号

2、此时选择相应选项会进入下一级菜单，例如选择1，回车：

请输入起始地：

沈阳

请输入目的地：

大连

车次：T11

起始地----->目的地： 沈阳----->大连

到达时间：2008 08 08 18 18

票价：88

3、购买车票

请输入您的姓名：

郑晨贾若

请输入起始地:

沈阳

请输入目的地:

大连

车次: T11

起始地-----> 目的地: 沈阳-----> 大连

到达时间: 2008 08 08 18 18

票价: 88

您确定购买此车票吗? 1.是 2.否

1

请输入您要购买的张数:

18

购买成功! 返回菜单

#### 4、退订车票

您确定要退票吗? 1.是 2.否

退票成功! 返回菜单

## 六、思考题

- 1、如何用指针调用结构体数组?
- 2、票务信息管理还有哪些可扩充的功能?
- 3、如何实现对车票、站名等信息的模糊查询?

# 第十章 超级终端仿真

本章要求设计一个串口通信程序，利用串口通信原理，实现对超级中端程序模拟。具体实现传送字符、接收字符的功能。（有关端口地址及相关命令请参考相关资料）

## 一、实践目的

- 1、理解串口通信原理，了解 RS—232 串行通信标准。
- 2、熟悉 TC 下对端口的控制操作，掌握各个端口地址值及控制数值。
- 3、熟悉 TC 编程环境，掌握编程方法。

## 二、基本要求

- 1、掌握 RS—232 串口硬件构成及相关知识。
- 2、了解 TC “dos.h”中 outportb() ,inportb(), setvect()等函数使用方法。
- 3、理解中断概念，掌握 TC 下实现中断过程。
- 4、系统制作完成后应实现类似图 10.1 所示界面。

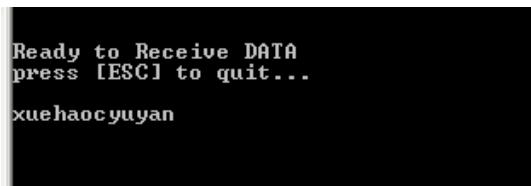


图 10.1 程序运行界面

## 三、算法分析

### 1、数据结构

设计一个存储发送接收字符的字符数组 unsigned char Buffer[BUFFLEN];

### 2、函数定义

程序源文件中具体包含的函数如表 10.1 所示。

表 10.1 文件及函数说明

源文件	函数原型	功能说明
super.c	void InitCOM()	初始化端口
	void OpenPort()	打开端口
	void ClosePort()	关闭端口
	void interrupt far asyncint(void)	设置中断服务程序
	void interrupt(*asyncoldvect)()	保存旧中断向量

### 3、处理过程

- （1）首先，使用 openport()打开端口，包括使用 InitCOM()对端口的初始化，在屏幕显示提

示信息，要求用户输入字符；使用 `kbhit()`判断键盘是否有输入，若有则读取并判断是否是 ESC 键，是则退出；否则使用 `sendchar()`函数将其发送。使用 `readchar()`接收来自端口的字符，若不为空，则显示在屏幕上，使用 `closeport()`关闭端口。

(2) 设计相关函数：

- 1) 初始化端口函数 `void InitCOM()`

```
outportb(0x3fb,0x80);//设置端口波特率
outportb(0x3f8,0x0c);//设置低位
outportb(0x3f9,0x00);//设置高位
outportb(0x3fb,0x03);//设置线路控制寄存器，数据长度8位
outportb(0x3fb,0x08|0x0b);//设置校验位
outportb(0x3f9,0x01);//允许接收器中断
}
```
- 2) 打开端口函数 `void OpenPort()`

```
unsigned char ucTemp;
InitCOM();
asyncoldvect=getvect(0x0c);//获取中断向量
disable();//禁止中断
inportb(0x3f8);//发送允许寄存器
inportb(0x3fb);//线路控制寄存器
inportb(0x3fa);//中断识别寄存器
outportb(0x3fb,0x08|0x0b);// 设置线路控制寄存器校验位
outportb(0x3f9,0x01);//中断允许寄存器允许接收器中断
ucTemp=inportb(0x21)&0xef;//字符编码加密
outportb(0x21,ucTemp);//向端口发送字符
setvect(0x0c,asyncint);//使用新的中断向量
enable();//允许中断
}
```
- 3) 关闭端口函数 `void ClosePort()`

```
disable();//禁止中断
outportb(0x3f9,0x00);//禁止接收器中断
outportb(0x3fb,0x00);//禁止线路寄存器更改
outportb(0x21,inportb(0x21)&0x10);//从端口读取字符并解码
enable();//允许中断
setvect(0x0c,asyncoldvect);//将中断向量恢复
}
```
- 4) `void interrupt far asyncint(void)`

```
Buffer[buffin++]=inportb(0x3f8);//从接收寄存其读取字符并存储
if(buffin>=BUFFLEN)
buffin=0;//若存储空间满则从头存储
outportb(0x20,0x20);//COM1口
}
```
- 5) 设置中断向量 `void interrupt(*asyncoldvect)()`//将旧中断向量保存在 `asyncoldvect` 中

## 四、基本步骤

### 1、打开 TC2.01 开发环境

双击 TC.exe 进入主程序界面，如图 10.2 所示。

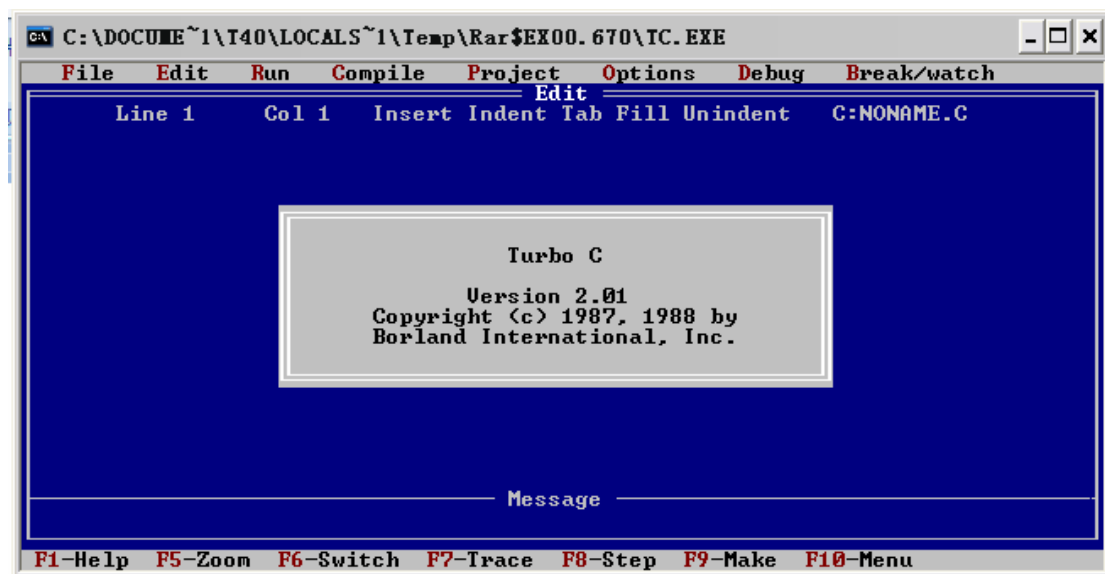


图 10.2 TC2.01 运行主界面

## 2、输入及导入代码

因 TC 不支持鼠标，因此建议打开“记事本”，在其中输入代码，保存为“chuankou.c”文件，并保存在 TC 安装目录下。另外也可直接在 TC 下输入代码。

同时按 Alt+F 选择菜单栏的【File】 | 【pick】，在 Loadfile 中选择\*.c 文件。在查找结果中选择 chuankou.c 文件,按回车确定。此时代码即可导入。

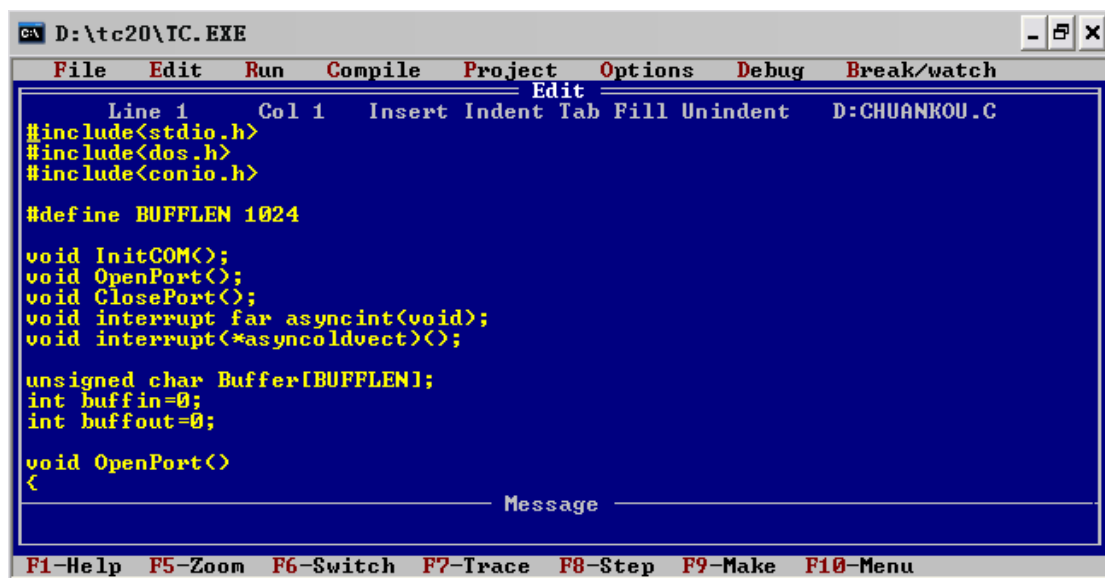


图 10.3 TC2.01 导入代码后界面

## 3、编译及调试

同时按 Alt+C 选择【Compile】编译运行。若无输入错误，则可出现图 10.4 所示界面，若此时串口硬件安装完毕，则可进行传输。



图 10.3 编译成功后界面

若出现错误，则出现 10.4 界面。

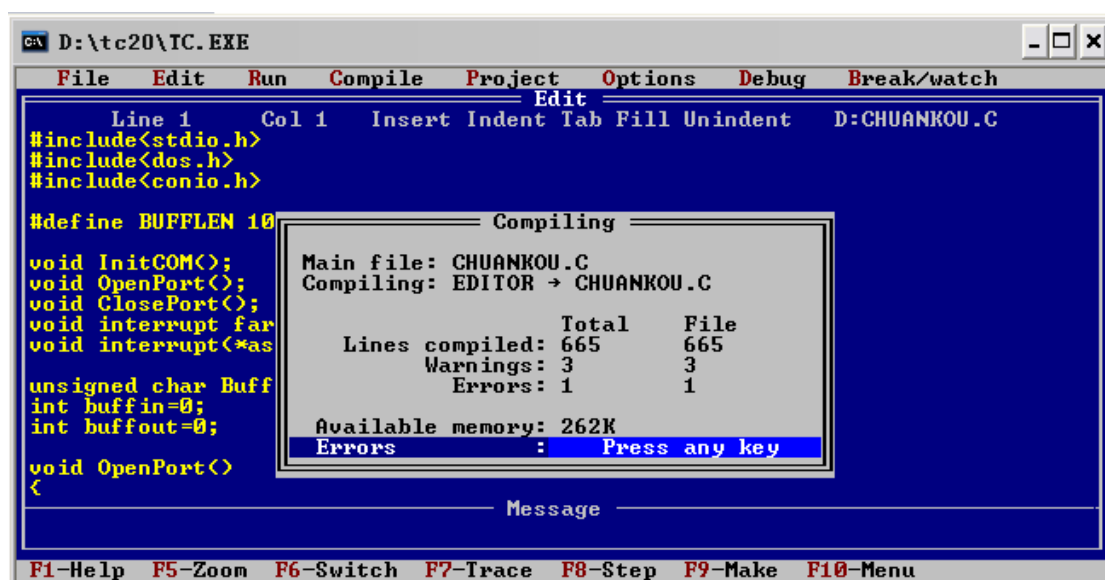


图 10.4 编译错误界面

退出 TC 在记事本中进行修改，之后重新导入即可，也可直接在 TC 下修改。修改时下方 Message 处显示错误情况，上方代码中相应错误则以高亮显示。使用键盘即可修改。注意修改之后要按 F2 保存，如图 10.5 所示。

#### 4、执行程序

编译无问题后，可按 Alt+R 运行，若无错误，即可出现图 10.1 界面。若此时串口硬件安装完毕，则可进行传输；否则继续修改，直到正确为止。



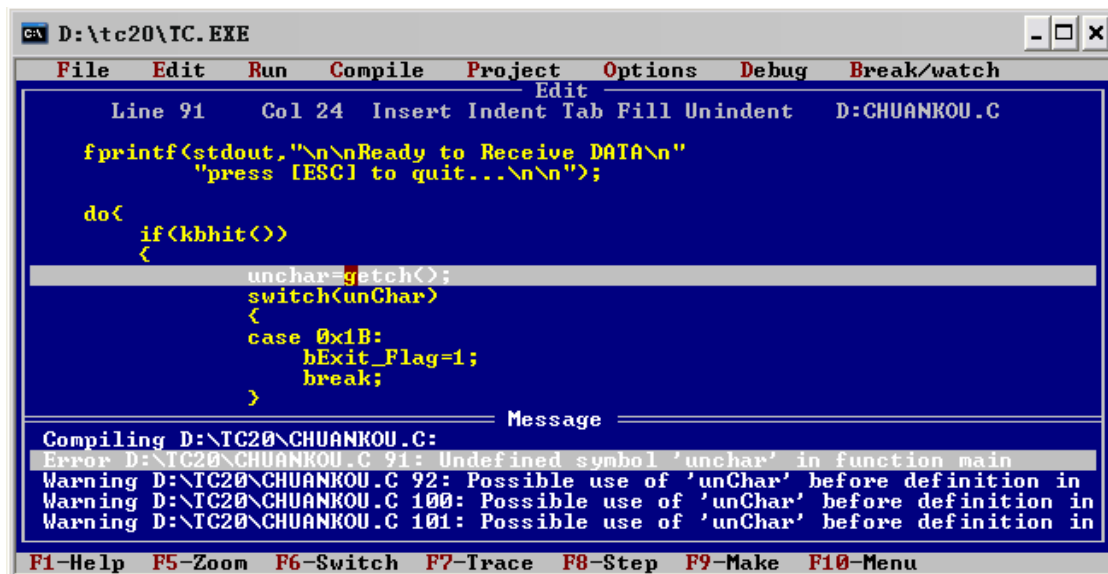


图 10.5 调试界面

## 五、参考代码

```

/*chuankou.c*/
#include<stdio.h>
#include<dos.h>
#include<conio.h>
#define BUFFLEN 1024
void InitCOM();
void OpenPort();
void ClosePort();
void interrupt far asyncint(void);
void interrupt(*asyncoldvect)();
unsigned char Buffer[BUFFLEN];
int buffin=0;
int buffout=0;

void OpenPort(){
    unsigned char ucTemp;
    InitCOM();
    asyncoldvect=getvect(0x0c);
    disable();
    inportb(0x3f8);
    inportb(0x3fe);
    inportb(0x3fb);
    inportb(0x3fa);
    outportb(0x3fb,0x08|0x0b);/*0x3fc改为x3fb为线路控制端口*/
    outportb(0x3f9,0x01);
    ucTemp=inportb(0x21)&0xef;
    outportb(0x21,ucTemp);
    setvect(0x0c,asyncint);
    enable();
}

void interrupt far asyncint(){
    Buffer[buffin++]=inportb(0x3f8);
    if(buffin>=BUFFLEN)

```

```

        buffin=0;
        outportb(0x20,0x20);
    }

void ClosePort(void){
    disable();
    outportb(0x3f9,0x00);
    outportb(0x3fb,0x00);/*0x3fc改为x3fb为线路控制端口*/
    outportb(0x21,inportb(0x21)&0x10);
    enable();
    setvect(0x0c,asyncoldvect);
}

void InitCOM(){
    outportb(0x3fb,0x80);
    outportb(0x3f8,0x0c);
    outportb(0x3f9,0x00);
    outportb(0x3fb,0x03);
    outportb(0x3fb,0x08|0x0b);/*0x3fc改为x3fb为线路控制端口*/
    outportb(0x3f9,0x01);
}

unsigned char read_char(void){
    unsigned unch;
    if(buffout!=buffin)
    {
        unch=Buffer[buffout];
        buffout++;
        if(buffout>=BUFFLEN)
            buffout=0;
        return(unch);
    }
    else
        return(0xff);
}

void send_char(unsigned char unch){
    while(((inp(0x3f8+5))&0x40)==0);
    outportb(0x3f8,unch);
}

void main(){
    unsigned char unChar;
    short bExit_Flag=0;
    OpenPort();
    fprintf(stdout,"\n\nReady to Receive DATA\n"
            "press [ESC] to quit...\n\n");
    do{
        if(kbhit()){
            unChar=getch();
            switch(unChar){
                case 0x1B:
                    bExit_Flag=1;
                    break;
            }
            if(!bExit_Flag){
                fprintf(stdout,"%c",unChar);
                send_char(unChar);
            }
        }
    }while(1);
}

```

```

        }
    }
    unChar=read_char();
    if(unChar !=0xff){
        fprintf(stdout,"%c",unChar);
    }
}while(!bExit_Flag);
ClosePort();
}

```

## 六、代码测试

程序运行后，在界面输入字符，对方程序界面将显示输入内容，同时本地程序将回显字符串。

```

Ready to Receive Date
Press [ESC] to quit.....
abcdefg

```

## 七、思考题

- 1、中断函数是如何实现中断的？
- 2、TC 中是如何导入代码的？
- 3、串口通讯编程的关键步骤是什么？