****

C语言课程设计报告

题 目： 火车订票系统

学 院： 计算机科学技术学院

专 业： 计算机类Ⅱ

姓 名： 李坤璘

学 号： 2019202216

指导教师： 兰淑丽

2021年 6 月 3 日

目录

[第一章 前言 2](#_Toc74781016)

[1.1 课程设计目的 2](#_Toc74781017)

[1.2 设计领域范围 2](#_Toc74781018)

[1.3 研究方法 2](#_Toc74781019)

[1.4 预期结果及意义 2](#_Toc74781020)

[第二章 需求分析 2](#_Toc74781021)

[2.1 功能需求 2](#_Toc74781022)

[2.2 性能需求 3](#_Toc74781023)

[第三章 总体设计 4](#_Toc74781024)

[3.1 设计思路 4](#_Toc74781025)

[3.2 程序结构 4](#_Toc74781026)

[第四章 详细设计 5](#_Toc74781027)

[4.1 main函数 5](#_Toc74781028)

[4.2 menu函数 7](#_Toc74781029)

[4.4 modifytrain函数 12](#_Toc74781030)

[4.5 showtrain函数 17](#_Toc74781031)

[4.6 preserve函数 19](#_Toc74781032)

[4.7 showman函数 24](#_Toc74781033)

[4.8 searchtrain函数 26](#_Toc74781034)

[4.9 modifyman函数 30](#_Toc74781035)

[4.10 cloud函数 36](#_Toc74781036)

[4.11 read函数 40](#_Toc74781037)

[第五章 结论体会 43](#_Toc74781038)

# 第一章 前言

## 1.1 课程设计目的

随着社会向高效、快速的趋势发展，人们的出行日益频繁，但人们常常没有多余的时间去购票、退票、改签等。提前去售票处买票，人多又耽误时间，询问售票人员到目的地的火车有哪些，时间是几点，票价是多少，是否还有余票等信息，可能不会问的太详细，这样的流程繁琐且容易出错。互联网的发展为人们提供了一个很好的购票渠道，通过互联网可以更好的获取火车票的信息和进行订票退票等。

## 1.2 设计领域范围

管理系统、C语言基本操作、数据结构——链表。

## 1.3 研究方法

算法分析、寻找最优解、流程图分析、调试。

## 1.4 预期结果及意义

书写出一整套火车票订票系统，并成功运行达到相关功能要求。

# 第二章 需求分析

## 2.1 功能需求

2.1.1 添加火车票信息

添加火车票可以随时加入新的火车信息，对管理系统进行实时更新，保证了时效性和信息的准确性。

2.1.2 查询火车票信息

查询火车信息可以根据用户不同的需求，输入不同的关键词，查询到符合条件的相关火车信息，查询方式的不同满足了多元化大众的需求。

2.1.3 订票系统

管理系统的核心功能，用户根据现存的车票来选择自己想要的车票进行预订购买，写下自己的姓名和身份证信息，并且将购票信息保存下来，同时更新火车信息。

2.1.4 修改火车票信息

对火车票的相应车次的信息进行修改，可以进行信息更新或者错误修正，提高了程序的容错率。

2.1.5 修改购票人信息

对购票人的相应车次的信息进行修改，可以进行信息更新或者错误修正，提高了程序的容错率，同时增加了改签功能，为乘客提供了更多的选择方式。

2.1.6 显示火车票信息

让用户浏览所有的火车信息，包括车次号、始发站、终点站、出发时间、到达时间、票价、剩余票数等所有的信息，便于用户挑选合适的车次。在每个功能中都会涉及到显示火车票信息。

2.1.7 显示购票人信息

让用户浏览自己已经订过的车票，包括火车票信息加上自己购买的票数、花费价格。

2.1.8 从文件读取存档

读取现有的文件信息，为购票系统起到初始化作用。

2.1.9 同步信息到文件

将已经存有的信息保存到文件中，以便下次使用的时候再重新读取使用。

## 2.2 性能需求

2.2.1 便于使用

面对于用户的黑框程序从理论上说是不友好不方便的，因此要加上足够多的提示信息，以及在系统中增添若干个回退功能，保证满足客户的所有需求，使用用户熟悉的术语还有中文界面，保证了程序的易用。

2.2.2 信息精确定位

对于链表遍历的时候保证所有的数据都是可用存在的，能够精确找到相应的位置，避免出现信息歧义的问题。

2.2.3避免重复信息

在每一个功能中都增添一个查重判断(由于不同功能查重方式不同所以没有写特定的查重函数)。

# 第三章 总体设计

## 3.1 设计思路

使用两个结构体分别存储火车和购票人的而信息，并用链表来储存。所有的功能都封装在相应的void类型的而函数，保证了系统的可读性良好。

在主函数里根据输入的选项来调用相应的函数，执行相关的功能。

将所有的信息存储在两个txt文件中，便于信息存储和下一次使用程序能够读取。

## 3.2 程序结构

本程序使用的是DEV C++开发环境，因此创建了一个DEV工程，内附属了1个自定义和11个.c项目文件，具体文件说明如下：

火车订票系统.dev——项目文件，用于将多个文件联系在一起运行工程。

structure.h——储存链表信息、函数声明的头文件。

Cloud.c——存储了cloud函数，用于火车票和购票信息的保存。

showman.c——存储了showman函数，用于输出展示所有购票人信息。

read.c——存储了read函数，用于把文件中的信息读入链表。

modifytrain.c——存储了modifytrain函数，用于修改链表中已存在的火车信息。

menu.c——存储了menu函数，用于打印主界面，以便用户使用相关功能。

addtrain.c——存储了addtrain函数，用于向链表中添加火车信息。

main.c——存储了代码初始化的内容和main函数，控制着整个程序的运行。

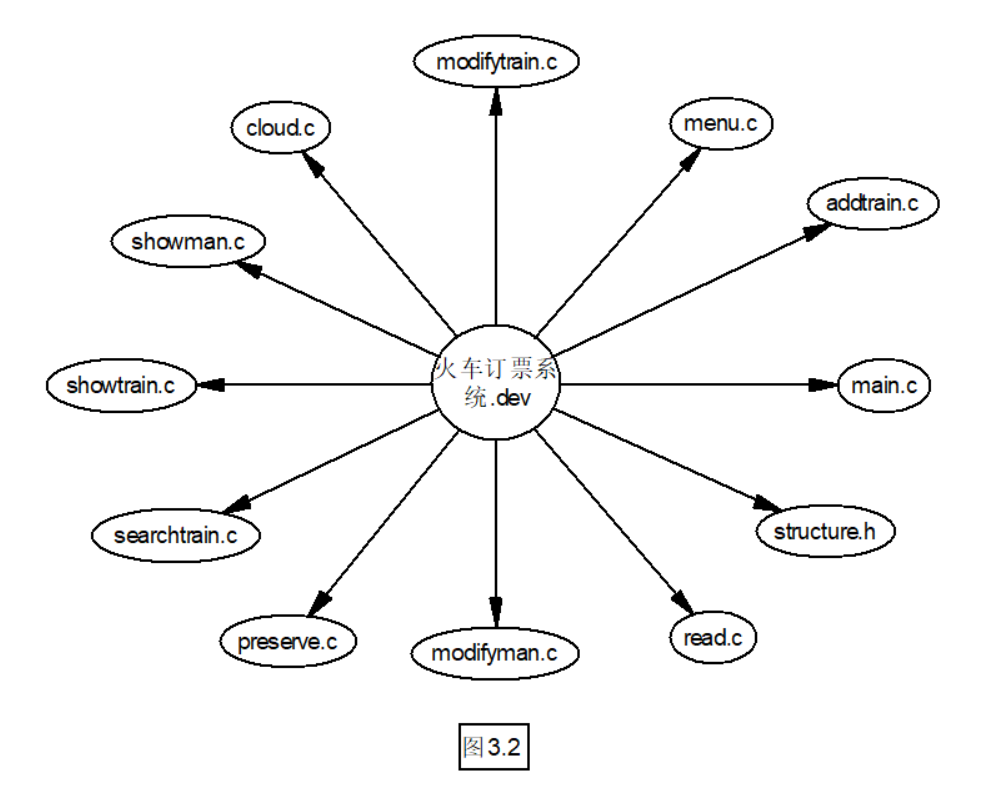
showtrain.c——存储了showtrain函数，用于在屏幕上打印链表中现有的信息。

searchtrain.c——存储了searchtrain函数，用于用户进行车次的查询。

preserve.c——存储了preserve函数，是用户的订票系统。

modifyman.c——存储了modifyman函数，用于修改链表中已存在的购票信息。

结构图如下：



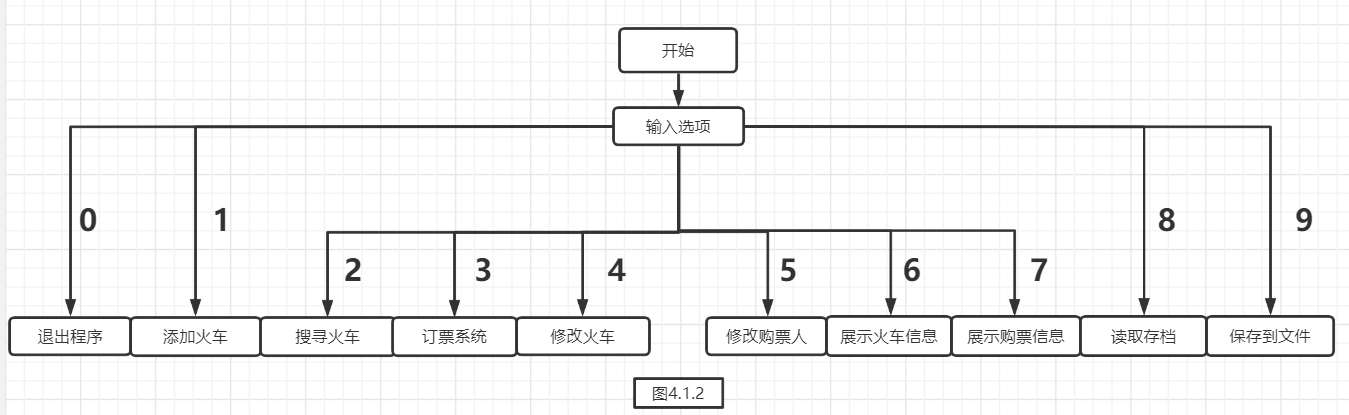
# 第四章 详细设计

## 4.1 main函数

4.1.1 函数功能

系统的主函数，控制调用相应的函数。

4.1.2 流程图



4.1.3 代码实现

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// main.c

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <stdio.h>

#include "structure.h"

#include <stdlib.h>

#include <string.h>

int main(void)

{

char n[10];

int flag = 0, \*a;

a = &flag;

node \*head = NULL; //建立火车链表的头部指针

book \*Head = NULL; //建立人火车链表的头指针

head = (node\*)malloc(sizeof(node)); Head = (book\*)malloc(sizeof(book));

head->next = 0; Head->next = NULL;

Head->next = NULL;

while (1) {

flag = 0;

menu();

printf("请输入0~9的数字选择操作内容：");

scanf("%s", n);

system("cls");

//printf("\n");

if(strcmp(n,"0")==0)

{

printf("感谢您的使用！\nThanks！\n");

system("pause");

return 0;

}

else if(strcmp(n,"1")==0) addtrain(head);

else if(strcmp(n,"2")==0) searchtrain(head);

else if(strcmp(n,"3")==0) preserve(head, Head);

else if(strcmp(n,"4")==0) modifytrain(head);

else if(strcmp(n,"5")==0) modifyman(head,Head);

else if(strcmp(n,"6")==0) showtrain(head, a);

else if(strcmp(n,"7")==0) showman(Head, a);

else if(strcmp(n,"8")==0) read(head,Head);

else if(strcmp(n,"9")==0) cloud(head, Head);

else printf("请检查信息是否正确！\n");

system("pause"); system("cls");

}

}

4.1.4 测试

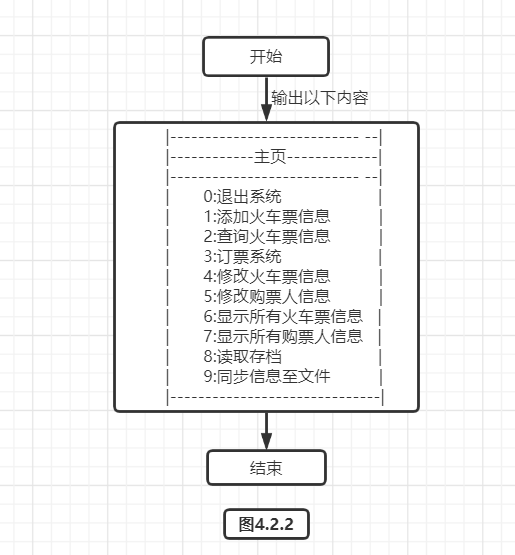
详见每一个功能函数的操作。

## 4.2 menu函数

4.2.1 函数功能

菜单栏，以便用户了解相关功能

4.2.2 流程图



4.2.3 代码实现

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// menu.c

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include<string.h>

#include<stdio.h>

void menu()

{

puts("\n");

puts("\t|-----------------------------|");

puts("\t|------------主页-------------|");

puts("\t|-----------------------------|");

puts("\t|\t0:退出系统 |");

puts("\t|\t1:添加火车票信息 |"); //批量添加

puts("\t|\t2:查询火车票信息 |"); //多种方式查询\*

puts("\t|\t3:订票系统 |"); //两个链表交互

puts("\t|\t4:修改火车票信息 |"); //序号查重特性\*

puts("\t|\t5:修改购票人信息 |"); //序号查重特性 和火车票链表交互\*

puts("\t|\t6:显示所有火车票信息 |");

puts("\t|\t7:显示所有购票人信息 |");

puts("\t|\t8:读取存档 |");

puts("\t|\t9:同步信息至文件 |");

puts("\t|-----------------------------|");

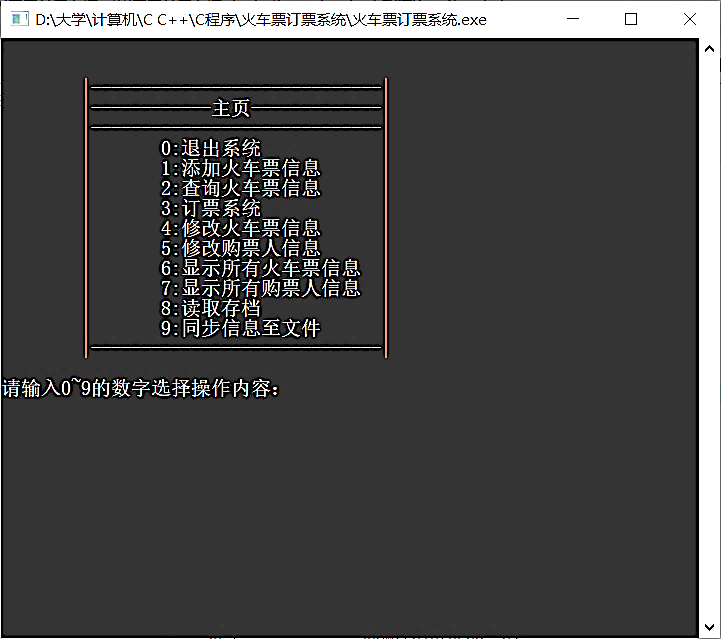
printf("\n");

}

//主函数用的字符串不是数字

//有goto语句辅助跳多种循环 还有回退操作

4.2.4 测试

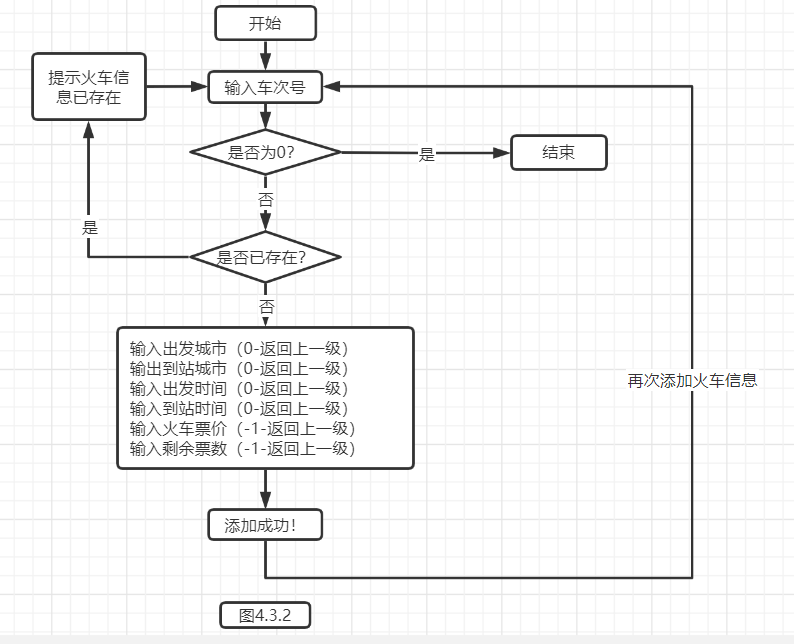


4.3 addtrain函数

4.3.1 函数功能

添加火车票信息，及时添加相关火车票的序号、出发地、到达地、时间、票价、票数，以便购票人能够查询到买到该车次火车票。

4.3.2 流程图



4.3.3 代码实现

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// addtrain.c

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

#include "structure.h"

void addtrain(node \*head)

{

node \*p,\*r,\*s;

char num[10];

r=head;

//遍历链表到尾部以便后续尾插

while(r->next!=NULL) r=r->next;

while(1)

{

kunkun://goto语句标记的地点

printf("请输入车次号(0-退出):");

scanf("%s",num);

if(strcmp(num,"0")==0)

{

printf("\n添加结束！\n"); break;

}

//每一次遍历前都将链表返回头部

s=head->next;

//遍历判断是否已存在

while(s)

{

if(strcmp(s->num,num)==0)

{

printf("车次号%s已存在，请重新输入！\n\n",num);

goto kunkun;//如果有重复的，就直接跳回最外循环的最初 “kunkun”

}//goto语句不提倡用，但在这里直接完成了break+continue两次操作

s=s->next;

}//\*\*\*一开始有关于嵌套循环用break/continue时出现了bug

//输入要插入的新信息

p=(node \*)malloc(sizeof(node));

strcpy(p->num,num);

printf("输入火车信息\n");

startcity:

printf("出发城市(0-返回上一级):");

scanf("%s",p->startcity);

if(strcmp(p->startcity,"0")==0)

{

printf("\n");

goto kunkun;

}

reachcity:

printf("到站城市(0-返回上一级):");

scanf("%s",p->reachcity);

if(strcmp(p->reachcity,"0")==0)

{

printf("\n"); goto startcity;

}

starttime:

printf("出发时间(0-返回上一级):");

scanf("%s",p->starttime);

if(strcmp(p->starttime,"0")==0)

{

printf("\n"); goto reachcity;

}

reachtime:

printf("到站时间(0-返回上一级):");

scanf("%s",p->reachtime);

if(strcmp(p->reachtime,"0")==0)

{

printf("\n"); goto starttime;

}

price:

printf("火车票价(-1-返回上一级):");

scanf("%d",&p->price);

if(p->price==-1)

{

printf("\n"); goto reachtime;

}

if(p->price<0)

{

printf("火车票价不能为负数！\n\n"); goto price;

}

ticketnum:

printf("剩余票数(-1-返回上一级):");

scanf("%d",&p->ticketnum);

if(p->ticketnum==-1)

{

printf("\n"); goto price;

}

if(p->ticketnum<0)

{

printf("票数不能为负数！\n\n"); goto ticketnum;

}

//尾插法

p->next=NULL;

r->next=p;

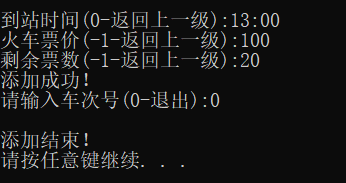
// r=r->next;

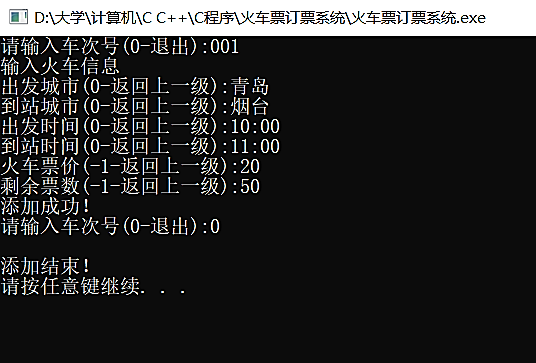
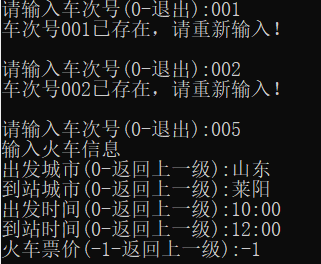
r=p;

printf("添加成功！\n");

}

}

4.3.4 测试

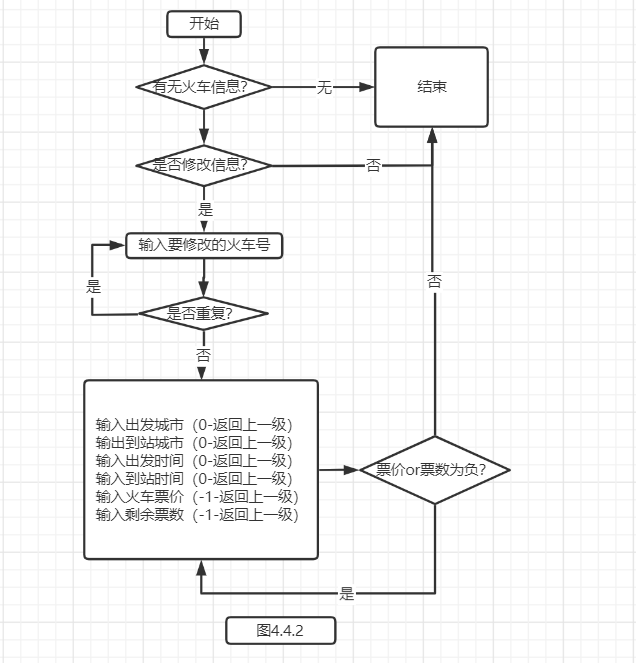


## 4.4 modifytrain函数

4.4.1 函数功能

修改现有的函数信息，以便乘务员因为导入信息错误进行及时的修改，提高了代码的容错率。

4.4.2 流程图



4.4.3 代码实现

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// modify.c

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <stdio.h>

#include "structure.h"

#include <string.h>

void showtrain(node \*head,int \*a);

void modifytrain(node \*head)

{

node \*p,\*q;

char num[10],ch[2],Num[10];

int flag=0,\*a;

a=&flag;

printf("目前所有的车票信息如下：\n");

showtrain(head,a);

if(flag) return ;

lili:

printf("您确定要修改信息吗？(y/n):");

//ch=getch();//吃掉

scanf("%s",ch);

if(strcmp(ch,"n")==0||strcmp(ch,"N")==0)

return ;

else if(strcmp(ch,"y")==0||strcmp(ch,"Y")==0)

{

while(1)

{

haha:

p=head->next;

printf("输入您要修改的车次号(0-退出):");

scanf("%s",num);

if(strcmp(num,"0")==0) return;

while(p!=NULL)

{

if(strcmp(p->num,num)==0)

{

printf("输入更改的火车信息\n");

Num:

//下面这段while(1)完美解决了如果修改的车次号和原来本有的车次号相同重复的问题！！！

while(1)

{

huhu:

printf("车次号(0-返回上一级)：");

scanf("%s",Num);

if(strcmp(num,"0")==0) goto haha;

q=head->next;

while(q!=NULL)

{

if(strcmp(Num,q->num)==0&&strcmp(Num,num)!=0)

{

printf("此车次号已存在！请重新输入！\n");

goto huhu;

}

q=q->next;

}

if(q==NULL)

{

strcpy(p->num,num);

break;

}

}

//上面这段while(1)完美解决了如果修改的车次号和原来本有的车次号相同重复的问题！！！

startcity:

printf("出发城市(0-返回上一级):");

scanf("%s",p->startcity);

if(strcmp(p->startcity,"0")==0)

{

printf("\n");

goto Num;

}

reachcity:

printf("到站城市(0-返回上一级):");

scanf("%s",p->reachcity);

if(strcmp(p->reachcity,"0")==0)

{

printf("\n");

goto startcity;

}

starttime:

printf("出发时间(0-返回上一级):");

scanf("%s",p->starttime);

if(strcmp(p->starttime,"0")==0)

{

printf("\n");

goto reachcity;

}

reachtime:

printf("到站时间(0-返回上一级):");

scanf("%s",p->reachtime);

if(strcmp(p->reachtime,"0")==0)

{

printf("\n");

goto starttime;

}

price:

printf("火车票价(-1-返回上一级):");

scanf("%d",&p->price);

if(p->price==-1)

{

printf("\n");

goto reachtime;

}

if(p->price<0)

{

printf("票价不能为负数！\n\n");

goto price;

}

ticketnum:

printf("火车票数(-1-返回上一级):");

scanf("%d",&p->ticketnum);

if(p->ticketnum==-1)

{

printf("\n");

goto price;

}

if(p->ticketnum<0)

{

printf("火车票数不能为负数！\n\n");

goto ticketnum;

}

printf("修改成功！\n");

break;

}

p=p->next;

}

if(p==NULL)//代表已经遍历到末尾了

{

printf("未查询到有关车次，请检查后重新输入。\n");

continue;

}

}

}

else

{

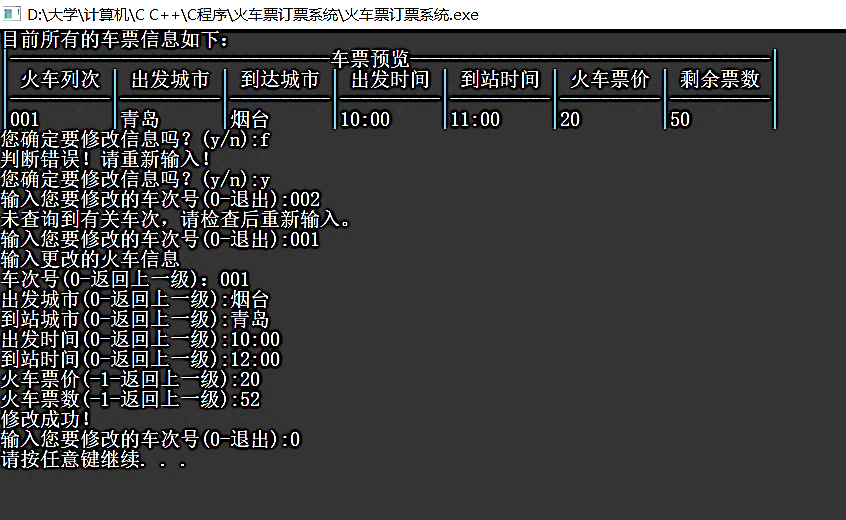
printf("判断错误！请重新输入！\n");

goto lili;

}

}

4.4.4 测试

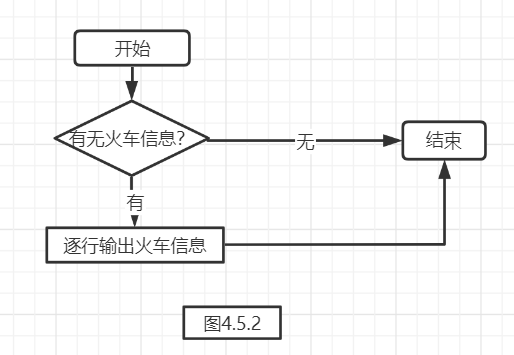


## 4.5 showtrain函数

4.5.1 函数功能

将所有的火车信息打印到屏幕中。

4.5.2 流程图



4.5.3 代码实现

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// showtrain.c

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <stdio.h>

#include "structure.h"

void showtrain(node \*head,int \*a)

{

node \*p;

p=head->next;

if(p==NULL)

{

printf("没有火车信息！！\n");

(\*a)=1;

}

else

{

printf("|--------------------------------车票预览------------------------------------|\n");

printf("| 火车列次 | 出发城市 | 到达城市 | 出发时间 | 到站时间 | 火车票价 | 剩余票数 |\n");

printf("|----------|----------|----------|----------|----------|----------|----------|\n");

while(p!=NULL)

{

printf("|%-10s|%-10s|%-10s|%-10s|%-10s|%-10d|%-10d|\n",p->num,p->startcity,p->reachcity,p->starttime,p->reachtime,p->price,p->ticketnum);

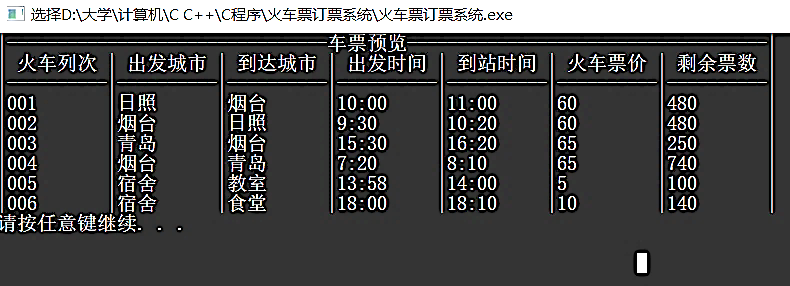
p=p->next;

}

}

}

4.5.4 测试

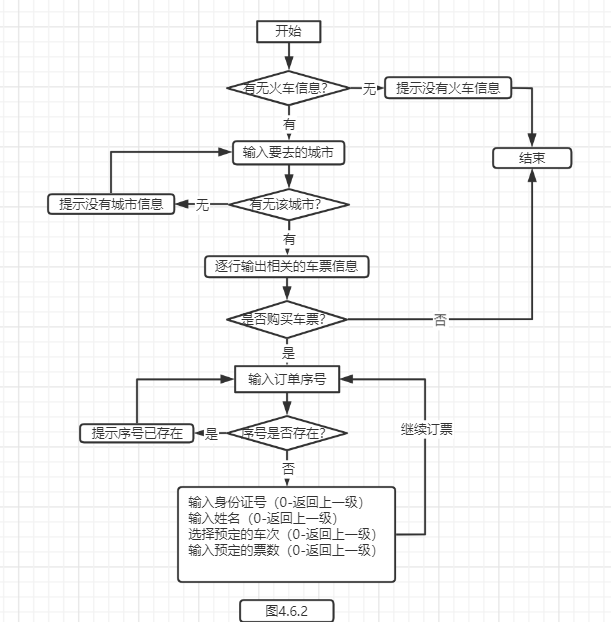


## 4.6 preserve函数

4.6.1 函数功能

最主要的函数，用户用此函数进行火车票的预订，接着存储相关的购票人信息。

4.6.2 流程图



4.6.3 代码实现

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// preserve.c

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <stdio.h>

#include <stdlib.h>

#include "structure.h"

#include <string.h>

void showtrain(node \*head,int \*a);

void preserve(node \*head,book \*Head)

{

node \*s[10];

char city[10],ch[2],Num[5];

int i=0,j,flag=0,\*a;

a=&flag;//flag为了判断火车链表是否为空的

node \*p=head->next;

book \*q=Head,\*ren,\*r=Head;

printf("目前所有的车票信息如下：\n");

showtrain(head,a);//调用showtrain展示所有火车

if(flag) return;//flag=1代表没有火车信息

while(1)

{

lalala:

i=0;

printf("输入要去的城市(0-退出)：");

scanf("%s",city);

if(strcmp(city,"0")==0)

return ;

p=head->next;//每次都要重新把链表放到头部来遍历

while(p!=NULL)

{

if(strcmp(city,p->reachcity)==0)

{

s[i]=p;i++;

}

p=p->next;

}

if(i==0)

{

printf("没有对应的火车票信息！请检查后重新输入\n\n");

goto lalala;

}

printf("相关的火车票如下\n");

printf("|------------------------车票预览------------------------------|\n");

printf("|火车列次|出发城市|到达城市|出发时间|到站时间|火车票价|剩余票数|\n");

printf("|--------|--------|--------|--------|--------|--------|--------|\n");

for(j=0;j<i;j++)

printf("|%-8s|%-8s|%-8s|%-8s|%-8s|%-8d|%-8d|\n",s[j]->num,s[j]->startcity,s[j]->reachcity,s[j]->starttime,s[j]->reachtime,s[j]->price,s[j]->ticketnum);

ning:

printf("您要预订车票吗？(y/n)(0-返回上一级)\n");

scanf("%s",ch);

if(strcmp(ch,"0")==0)

{

printf("\n");goto lalala;

}

if(strcmp(ch,"n")==0||strcmp(ch,"N")==0)

return ;

else if(strcmp(ch,"y")==0||strcmp(ch,"Y")==0)

{

aa:

printf("请输入人的信息：\n");

Num:

printf("订单序号(0-返回上一级)：");

scanf("%s",Num);

if(strcmp(Num,"0")==0)

{

printf("\n");goto ning;//58行

}

r=Head->next;

while(r!=NULL)

{

if(strcmp(r->Num,Num)==0)

{

printf("序号已存在！请重新输入！\n\n");goto aa;

}

r=r->next;

}

ren=(book \*)malloc(sizeof(book));//临时添加的人的结构体

strcpy(ren->Num,Num);

id:

printf("身份证号(0-返回上一级)：");

scanf("%s",ren->id);

if(strcmp(ren->id,"0")==0)

{

printf("\n");goto Num;

}

name:

printf("姓名(0-返回上一级)：");

scanf("%s",ren->name);

if(strcmp(ren->name,"0")==0)

{

printf("\n");goto id;

}

kun:

printf("选择要预定的车次(0-返回上一级)：");

scanf("%s",ren->num);

if(strcmp(ren->num,"0")==0)

{

printf("\n");goto name;

}

while(q->next!=NULL)

q=q->next;//把存储人信息的链表遍历到尾端

for(j=0;j<i;j++)

{

if(strcmp(ren->num,s[j]->num)==0)

{

booknum:

printf("请输入要预定的票数(-1-返回上一级)：");

scanf("%d",&ren->booknum);

if(ren->booknum==-1)

{

printf("\n");goto kun;

}

if(ren->booknum<0)

{

printf("票数不能为负数！\n\n");goto booknum;

}

if(s[j]->ticketnum - ren->booknum<0)

{

printf("票已不足！请重新选择\n\n");goto booknum;

}

s[j]->ticketnum -= ren->booknum;

strcpy(ren->startcity,s[j]->startcity);

strcpy(ren->reachcity,s[j]->reachcity);

strcpy(ren->starttime,s[j]->starttime);

strcpy(ren->reachtime,s[j]->reachtime);

ren->price=s[j]->price;

//人员链表尾插

ren->next=NULL;q->next=ren;

q=ren;break;

}

}

if(j==i)

{

printf("未查询到相关车次！请重新选择\n\n");goto kun;

}

//free(ren);

}

else

{

printf("判断错误！请重新输入！\n\n");

goto ning;

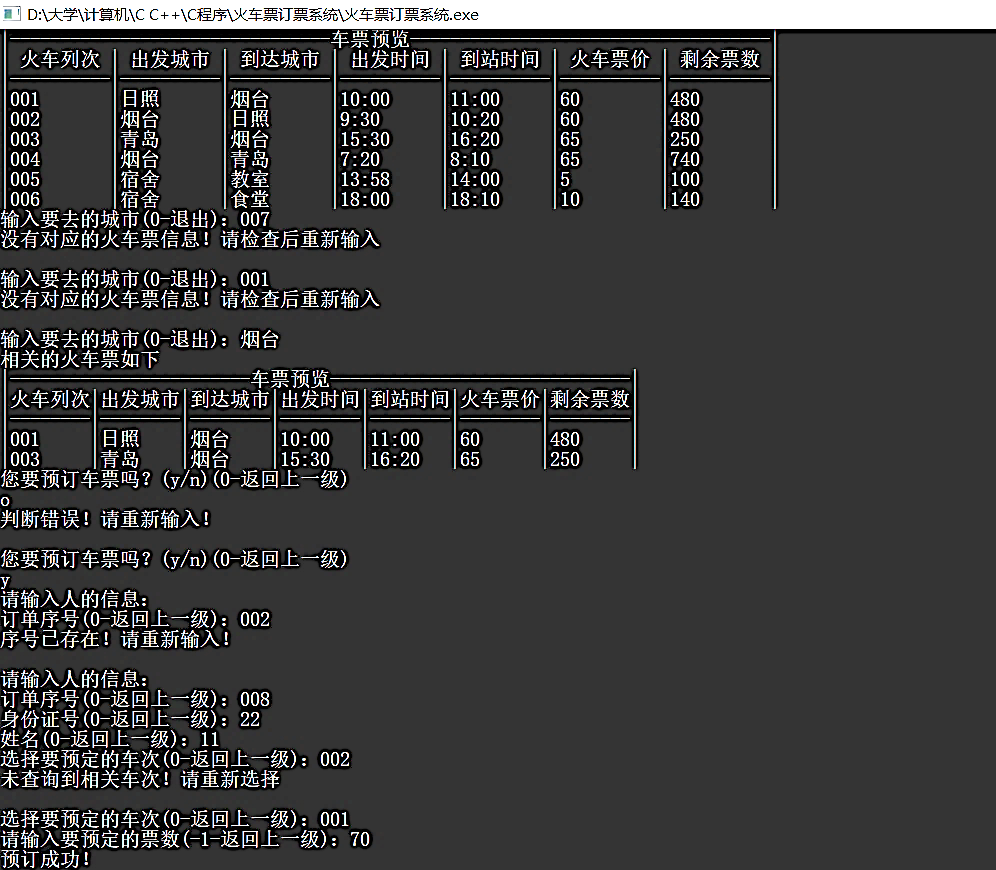
}

printf("预订成功！\n\n");

}

}

4.6.4 测试

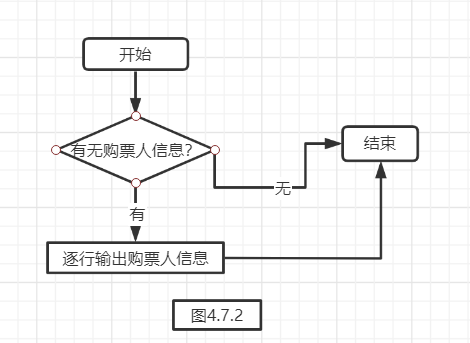


## 4.7 showman函数

4.7.1 函数功能

将所有的购票人信息打印到屏幕中，以便显示都有哪些人买了票，买了哪一站。

4.7.2 流程图



4.7.3 代码实现

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// showman.c

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <stdio.h>

#include "structure.h"

void showman(book \*Head,int \*a)

{

book \*p;

p=Head->next;

if(p==NULL)

{

printf("没有购票人信息！！\n");

(\*a)=1;

}

else

{

printf("|--------------------------------------购票人员预览---------------------------------------------------|\n");

printf("|车票序号| 身份证号 | 姓名 |火车列次|出发城市|到达城市|出发时间|到站时间|火车票价|购买票数|总共花费|\n");

printf("|--------|-----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|\n");

while(p!=NULL)

{

printf("|%-8s|%-11s|%-8s|%-8s|%-8s|%-8s|%-8s|%-8s|%-8d|%-8d|%-8d|\n",p->Num,p->id,p->name,p->num,p->startcity,p->reachcity,p->starttime,p->reachtime,p->price,p->booknum,p->price\*p->booknum);

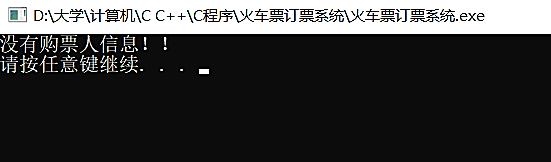
p=p->next;

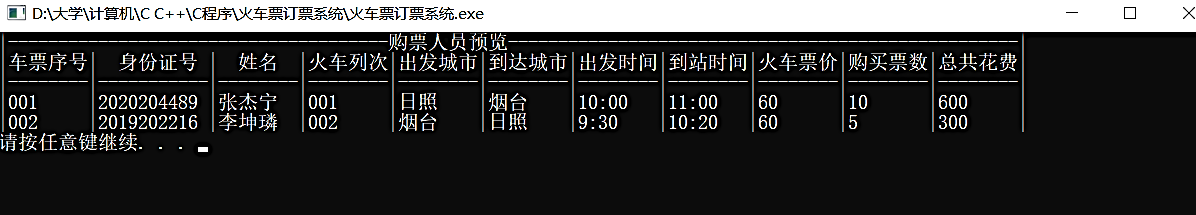
}

}

}

4.7.4 测试



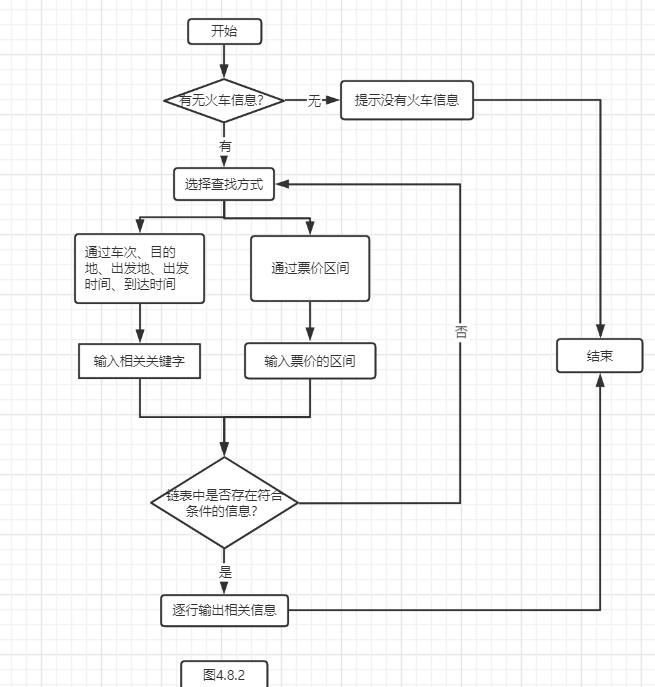


## 4.8 searchtrain函数

4.8.1 函数功能

根据不同的关键词/条件，筛选查询出想要查到的火车信息。

4.8.2 流程图



4.8.3 代码实现

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// searchtrain.c

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <stdio.h>

#include <string.h>

#include "structure.h"

void searchtrain(node \*head)

{

int n,a,b,i,j;

node \*p,\*s[10];

char num[5],other[10],tem[10];

if(head->next==NULL)

{

printf("没有火车信息！\n");

return ;

}

printf("1.通过车次 2.通过出发地\n");

printf("3.通过目的地 4.通过出发时间\n");

printf("5.通过到达时间 6.通过票价区间\n");

while(1)

{

printf("请选择您的查找方式(0-退出)：");

p=head->next;

scanf("%d",&n);

if(n==0)

{

printf("查询结束！\n");

return ;

}

else if(n==1)

{

printf("请输入车次号：");

scanf("%s",num);

while(p!=NULL)

{

if(strcmp(num,p->num)==0)

{

printf("|------------------------车票预览------------------------------|\n");

printf("|火车列次|出发城市|到达城市|出发时间|到站时间|火车票价|剩余票数|\n");

printf("|--------|--------|--------|--------|--------|--------|--------|\n");

printf("|%-8s|%-8s|%-8s|%-8s|%-8s|%-8d|%-8d|\n",p->num,p->startcity,p->reachcity,p->starttime,p->reachtime,p->price,p->ticketnum);

break;

}

p=p->next;

}

if(p==NULL)

printf("未查询到相关车次！\n");

}

else if(n>=2&&n<=5)

{

i=0;

printf("请输入相关信息：");

scanf("%s",other);

while(p!=NULL)

{

switch(n)

{

case 2:

strcpy(tem,p->startcity);break;//对比和下面的区别

//tem=p->startcity;break;

case 3:

strcpy(tem,p->reachcity);break;

case 4:

strcpy(tem,p->starttime);break;

case 5:

strcpy(tem,p->reachtime);break;

}

if(strcmp(other,tem)==0)

{

s[i]=p;

i++;

}

p=p->next;

}

if(i==0)

printf("未查询到相关车次！\n");

else

{

printf("|------------------------车票预览------------------------------|\n");

printf("|火车列次|出发城市|到达城市|出发时间|到站时间|火车票价|剩余票数|\n");

printf("|--------|--------|--------|--------|--------|--------|--------|\n");

for(j=0;j<i;j++)

printf("|%-8s|%-8s|%-8s|%-8s|%-8s|%-8d|%-8d|\n",s[j]->num,s[j]->startcity,s[j]->reachcity,s[j]->starttime,s[j]->reachtime,s[j]->price,s[j]->ticketnum);

}

}

else if(n==6)

{

linlin:

i=0;

printf("输入票价的区间：[a,b]：");

scanf("%d%d",&a,&b);

if(a>b)

{

printf("格式错误！请重新输入!\n\n");

goto linlin;

}

printf("票价的区间为[%d,%d]\n",a,b);

p=head->next;

while(p!=NULL)

{

if(p->price>=a&&p->price<=b)

{

s[i]=p;

i++;

}

p=p->next;

}

if(i==0)

printf("未查询到相关车次！\n");

else

{

printf("|------------------------车票预览------------------------------|\n");

printf("|火车列次|出发城市|到达城市|出发时间|到站时间|火车票价|剩余票数|\n");

printf("|--------|--------|--------|--------|--------|--------|--------|\n");

for(j=0;j<i;j++)

printf("|%-8s|%-8s|%-8s|%-8s|%-8s|%-8d|%-8d|\n",s[j]->num,s[j]->startcity,s[j]->reachcity,s[j]->starttime,s[j]->reachtime,s[j]->price,s[j]->ticketnum);

}

}

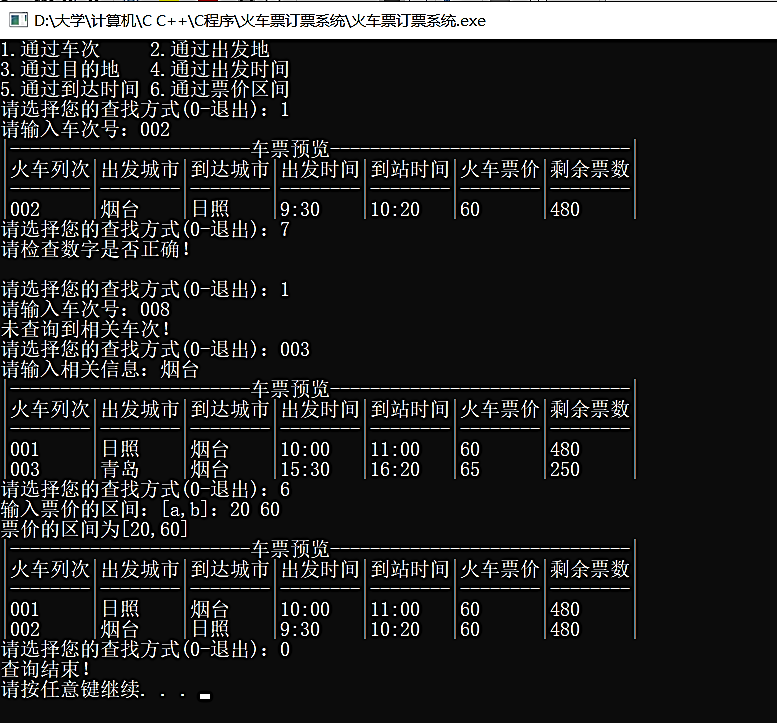
else

printf("请检查数字是否正确！\n\n");

}

}

4.8.4 测试



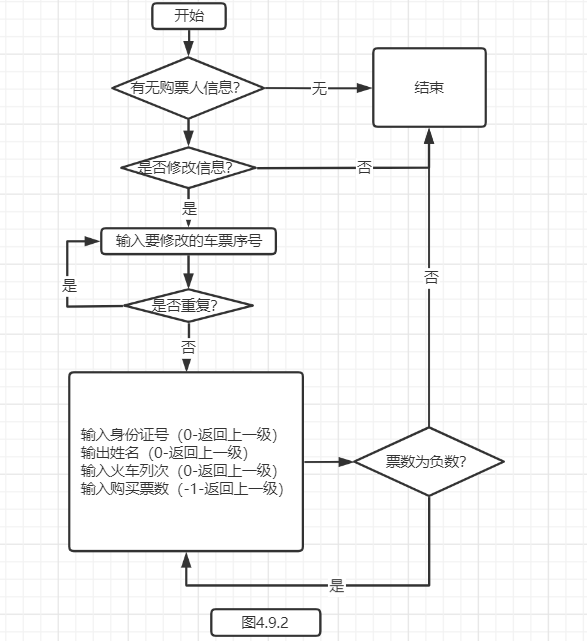
## 4.9 modifyman函数

4.9.1 函数功能

修改现有的函数信息，以便购票人因为导入信息错误进行及时的修改；

提高了代码的容错率，同时也可以进行退票和改签的操作。

4.9.2 流程图



4.9.3 代码实现

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// modifyman.c

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <stdio.h>

#include "structure.h"

#include <string.h>

void showman(book \*Head,int \*a);

void showtrain(node \*head,int \*a);

void modifyman(node \*head,book \*Head)

{

book \*p,\*q;

node \*r;

char num[10],ch[2],Num[10],NUM[10],STARTCITY[10],REACHCITY[10],STARTTIME[10],REACHTIME[10];

int flag=0,\*a,booknum,old\_booknum,PRICE;

a=&flag;

printf("目前所有的乘车人信息如下：\n");

showman(Head,a);

if(flag) return ;

lili:

printf("您确定要修改信息吗？(y/n):");

scanf("%s",ch);

if(strcmp(ch,"n")==0||strcmp(ch,"N")==0)

return ;

else if(strcmp(ch,"y")==0||strcmp(ch,"Y")==0)

{

while(1)

{

haha:

p=Head->next;

printf("输入您要修改的车票序号(0-退出):");

scanf("%s",num);

if(strcmp(num,"0")==0) return;

while(p!=NULL)

{

if(strcmp(p->Num,num)==0)

{

printf("输入更改的火车信息\n");

//下面这段while(1)完美解决了如果修改的车票序号和原来本有的车票序号相同重复的问题！！！

while(1)

{

huhu:

printf("车票序号(0-返回上一级)：");

scanf("%s",Num);

if(strcmp(Num,"0")==0)

{

printf("\n");

goto haha;

}

q=Head->next;

while(q!=NULL)

{

if(strcmp(Num,q->Num)==0&&strcmp(Num,num)!=0)

{

printf("此车票序号已存在！请重新输入！\n");

goto huhu;

}

q=q->next;

}

if(q==NULL)//说明想修改的车票序号和历史版本没有冲突

{

strcpy(p->Num,num);

break;

}

}

//上面这段while(1)完美解决了如果修改的车票序号和原来本有的车票序号相同重复的问题！！！

char ID[20];

id:

printf("身份证号(0-返回上一级):");

scanf("%s",ID);

if(strcmp(ID,"0")==0)

{

printf("\n");

goto huhu;

}

char NAME[10];

name:

printf("姓名(0-返回上一级):");

scanf("%s",NAME);

if(strcmp(NAME,"0")==0)

{

printf("\n");

goto id;

}

printf("目前所有可修改成的车票信息如下：\n");

showtrain(head,a);

if(flag) return ;

hhhhh:

r=head->next;

printf("火车列次(0-返回上一级):");

scanf("%s",num);

if(strcmp(num,"0")==0)

{

printf("\n");

goto name;

}

while(r!=NULL)

{

if(strcmp(r->num,num)==0)

{

strcpy(NUM,num);

break;

}

r=r->next;

}

if(r==NULL)

{

printf("没有相关火车列次，请检查后重新输入！\n");

goto hhhhh;

}

strcpy(STARTCITY,r->startcity);

strcpy(REACHCITY,r->reachcity);

strcpy(STARTTIME,r->starttime);

strcpy(REACHTIME,r->reachtime);

PRICE=r->price;

printf("相关的出发城市、到达城市、出发时间、到站时间、火车票价均已同步完毕\n");

kun:

printf("购买票数(-1-返回上一级):");

scanf("%d",&booknum);

if(booknum<0)

{

if(booknum==-1)

{

printf("\n");

goto hhhhh;

}

printf("购买票数不能为负数！\n\n");

goto kun;

}

if(booknum>r->ticketnum+p->booknum)

{

printf("票已不足！请重新选择票数\n\n");

goto kun;

}

old\_booknum=p->booknum;//这是修改前定的票数

p->booknum=booknum;

r->ticketnum=r->ticketnum+old\_booknum-booknum;

strcpy(p->id,ID);

strcpy(p->name,NAME);

strcpy(p->num,NUM);

strcpy(p->startcity,STARTCITY);

strcpy(p->reachcity,REACHCITY);

strcpy(p->starttime,STARTTIME);

strcpy(p->reachtime,REACHTIME);

p->price=PRICE;

printf("修改成功！\n");

break;

}

p=p->next;

}

if(p==NULL)//代表已经遍历到末尾了

{

printf("未查询到有关购票信息，请检查后重新输入。\n\n");

continue;

}

}

}

else

{

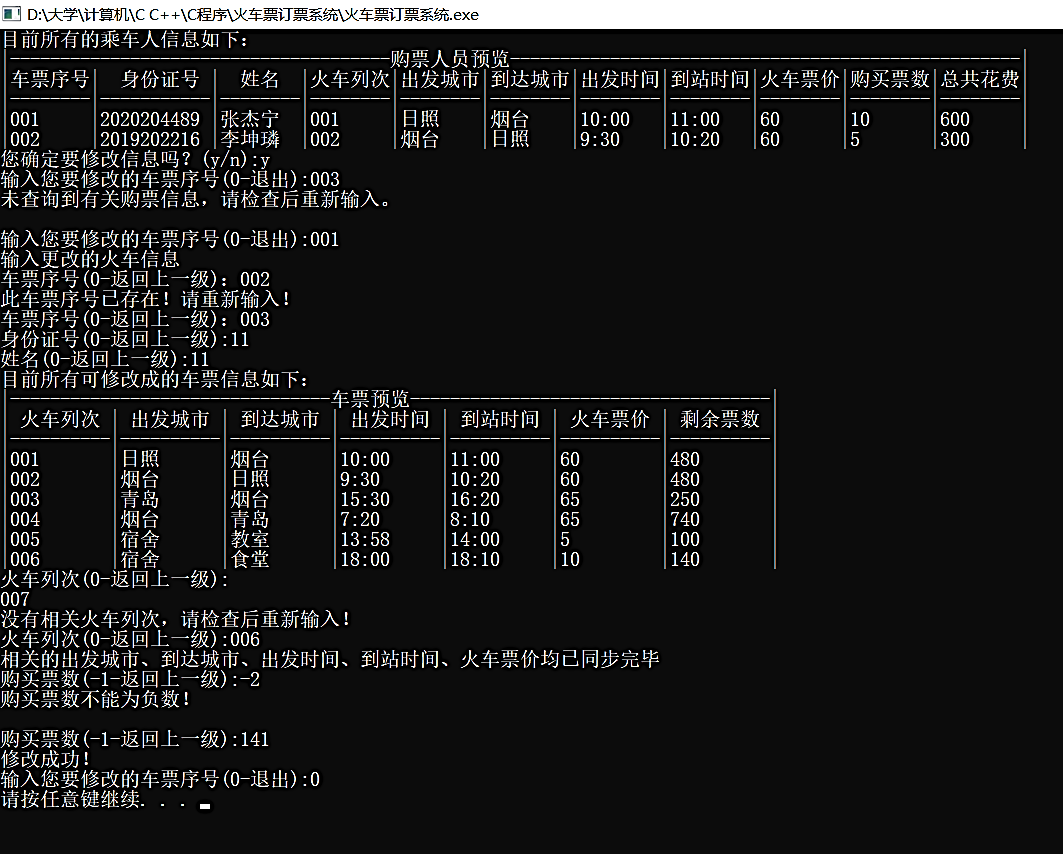
printf("判断错误！请检查后重新输入！\n\n");

goto lili;

}

}

4.9.4 测试

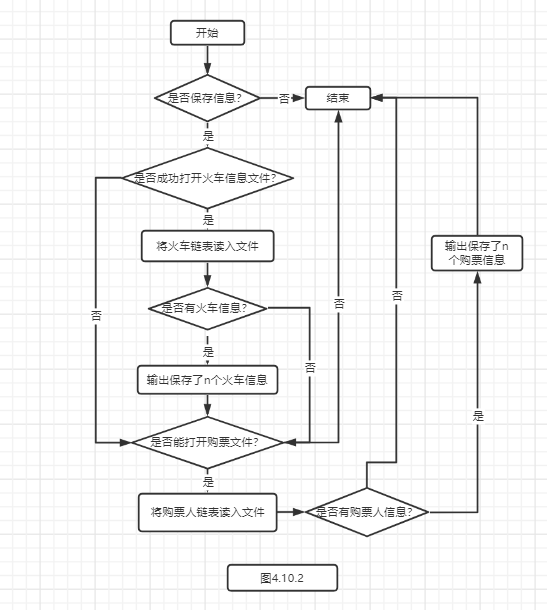


## 4.10 cloud函数

4.10.1 函数功能

将所有的购票人信息和火车票信息保存到文件中。

4.10.2 流程图



4.10.3 代码实现

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// cloud.c

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <stdio.h>

#include <string.h>

#include "structure.h"

#include <stdlib.h>

void cloud(node \*head,book \*Head)

{

char ch[10];

lili:

printf("您确定要同步保存信息吗？(y/n):");

//ch=getch();//吃掉

scanf("%s",ch);/////////////////////////////同样不知道为什么用ch有bug emmmmmm

if(strcmp(ch,"n")==0||strcmp(ch,"N")==0)

return ;

else if(strcmp(ch,"y")==0||strcmp(ch,"Y")==0)

{

FILE \*fp1,\*fp2;

node \*p;

int n=0;

p=head->next;

fp1=fopen("火车信息.txt","wb");

if(fp1==NULL)

{

printf("“火车信息”文件打开失败\n保存失败！\n");

goto lala;

}

if(p==NULL)

{

printf("没有火车信息！\n");

fclose(fp1);

goto lala ;

}

while(p!=NULL)

{

fwrite(p,sizeof(node),1,fp1);

n++;

p=p->next;

}

fclose(fp1);

printf("成功保存了%d个火车信息！\n",n);

lala:

n=0;

book \*q;

fp2=fopen("购票信息.txt","wb");

if(fp2==NULL)

{

printf("“购票信息”文件打开失败\n保存失败！\n");

if(fp1!=NULL)

fclose(fp1);

return ;

}

q=Head->next;

if(q==NULL)

{

printf("没有购票信息！\n");

return ;

}

while(q!=NULL)

{

fwrite(q,sizeof(book),1,fp2);

n++;

q=q->next;

}

fclose(fp2);

printf("成功保存了%d个购票信息！\n",n);

}

else

{

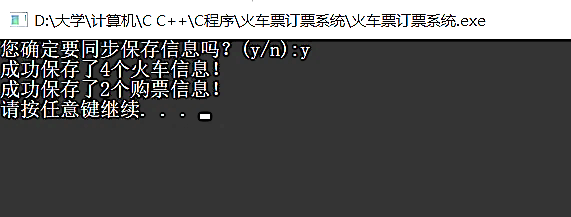
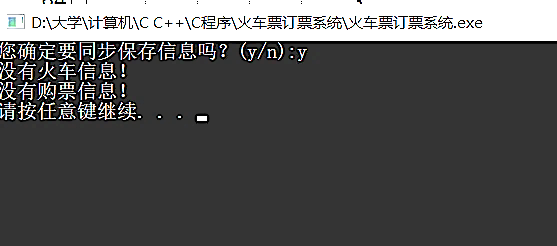
printf("判断错误！请检查后重新输入！\n");

goto lili;

}

}

4.10.4 测试

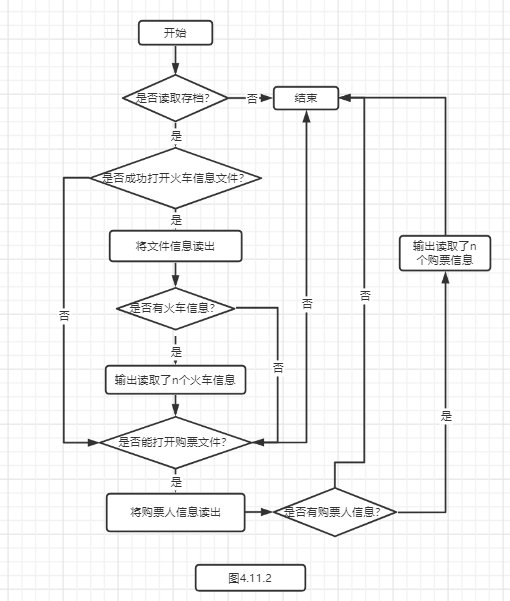


## 4.11 read函数

4.11.1 函数功能

将文件中的火车信息还有购票人信息读入各自链表中。

4.11.2 流程图



4.11.3 代码实现

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// read.c

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <stdio.h>

#include <string.h>

#include "structure.h"

#include <stdlib.h>

void read(node \*head,book \*Head)

{

node \*p=head,\*q;

char ch[10];

lili:

printf("您确定要读取存档吗？(y/n):");

//ch=getch();//吃掉

scanf("%s",ch);

if(strcmp(ch,"n")==0||strcmp(ch,"N")==0)

return ;

else if(strcmp(ch,"y")==0||strcmp(ch,"Y")==0)

{

int n=0;

FILE \*fp1;

fp1=fopen("火车信息.txt","rb");

if(fp1==NULL)

{

printf("“火车信息”文件打开失败\n读取失败！\n");

goto xiayibu;

}

while(!feof(fp1))

{

q=(node\*)malloc(sizeof(node));

if(fread(q,sizeof(node),1,fp1)==1)

{

q->next=NULL;

p->next=q;

p=q;

n++;

}

}

if(!n)

printf("未发现要导入的火车信息！\n");

else

printf("导入了%d个火车信息\n",n);

fclose(fp1);

xiayibu:

n=0;

FILE \*fp2;

fp2=fopen("购票信息.txt","rb");

if(fp2==NULL)

{

printf("“购票信息”文件打开失败\n读取失败！\n");

if(fp1!=NULL)

fclose(fp1);

return;

}

book \*m=Head,\*r;

while(!feof(fp2))

{

r=(book\*)malloc(sizeof(book));

if(fread(r,sizeof(book),1,fp2)==1)

{

r->next=NULL;

m->next=r;

m=r;

n++;

}

}

if(!n)

printf("未发现要导入的购票信息！\n");

else

printf("导入了%d个购票信息\n",n);

fclose(fp2);

}

else

{

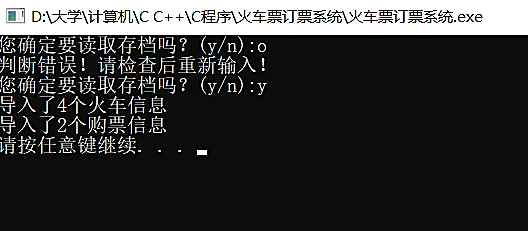
printf("判断错误！请检查后重新输入！\n");

goto lili;

}

}

4.11.4 测试



# 第五章 结论体会

经过四周的努力，这个火车票购票系统也算是拉下了帷幕。在之前几乎都没有想过自己可以那么快入手一个信息管理系统的设计，而且整个程序没有按照教材的步骤，是按照自己的思路和想法来运行的。虽然中途出现了许多bug，但是有bug就是代表自己的知识不熟练，没有熟练掌握，算是一种对自己的检测和进步。比如链表指向位置地方运行错误、文件操作fread返回值1才正确、输入流中会储存信息、字符串的判断、链表的交互等等东西都是之前不熟练的东西。通过这次程序设计，我知道了实践是最好的学习方式，在实践中提高了自己解决编程问题的能力。

其实这次的实验报告也充满了坎坷（笑哭），这次不仅仅学会了大程序的书写，还更加学会了课程报告的书写格式、文档排版以及图片编辑的操作，以及流程图的正确书写方式，其实这也算是对于之后的开发起到了一个好的开头，毕竟这些东西在之后用的只会越来越多，能够把代码写出来把程序运行出来是真功夫，但能够完整清晰剖析结构把它以报告形式呈现出来，这也算我们应该学会的。

在以后的日子里要加强锻炼自己的编程能力，不仅要熟练掌握课上所学，课下也要多多涉猎，提高自己的编程能力。不过，在之后的编程，我会变得越来越自信，越来越具有想法，在项目设计会取得自己该有的成绩。

感谢这几周努力的自己；

感谢同学们对于自己的帮助也启发；

也感谢老师，为我们的程序开发起到了启蒙作用；

在这条路上，以后走的一定会越来越远。