火车票务管理系统

目录

一、**需求分析..............................................2**

**二、概要设计..............................................2**

**三、详细设计..............................................3**

**四、调试分析..............................................6**

**五、用户手册..............................................7**

**六、测试数据..............................................7**

**七、附录...**..............................................10

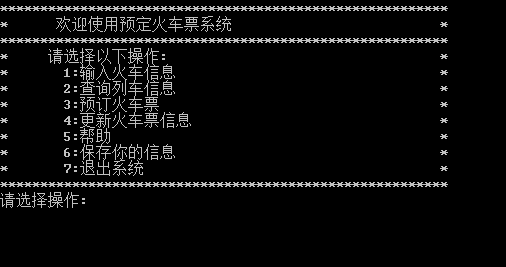
1. **需求分析**
2. 该程序具备火车票信息的输入、火车票信息的修改、查询、预定和退订等功能。程序编译环境为 VC6.0。
3. 火车票更新功能，用户先输入相关火车信息，程序将这些信息存入文件中。修改功能，用户输入火车信息后，根据提示可以修改该火车相关信息。查询功能，用户根据提示可查询需要的火车信息。用户可根据提示预定火车票，预定后输入自己的相关信息，即可预定成功。成功后可选择保存选项保存用户的信息。
4. **概要设计**
5. 整个系统中除了主函数外，还有各种功能的子函数，使用了while（）循环语句，switch（）调用语句，系统根据输入的数字选项来调用相应的函数，以实现相应的功能。主要实现了火车信息的输入、修改、及查询等功能。
6. void InsertTraininfo ( )：该函数实现了信息的输入，将火车的信息输入后并保存入文件中等功能。
7. void QueryTrain( )：该函数实现了信息的查询，它有一个int型的参数，参数为1时实现按I火车号查找火车信息，参数为2时实现按目的地查找火车信息，用户根据提示可查询所需要的火车票信息。
8. void BookTicket( )：该函数功能为预定火车票，通过输入火车号或目的地来预定火车票。
9. void UpdateInfo( )：该函数用于信息的修改更新，它主要实现了将文件里的火车信息读进链表中，通过查找，可将原来保存的信息进行修改并保存至文件中。
10. void printInterface( )：该函数它有一个int型的参数，参数为1时实现输入火车信息功能，参数为2时实现查找火车信息功能，参数为3时实现预定火车票功能，参数为4时实现更新火车信息功能，参数为5时实现提示帮助预定火车票功能。
11. 程序流程：

main

printInterface

InsertTraininfo QueryTrain BookTicket UpdateInfo

1. 主界面如下：



三、**详细设计**

1 主要数据结构

struct train

{

char num[10];/\*列车号\*/

char city[10];/\*目的城市\*/

char takeoffTime[10];/\*发车时间\*/

char receiveTime[10];/\*到达时间\*/

int price;/\*票价\*/

int bookNum ;/\*预定票数\*/

};

struct man

{

char num[10];/\*ID\*/

char name[10];/\*姓名\*/

int bookNum ;/\*需要的票数\*/

};

2 main（）函数：

选择

输入 查询 预订 更新 帮助 保存 退出

输入

输入

输入火车号 目的地

修改

火车票 目的地

输入你的信息

预定成功

3 void InsertTraininfo ( )函数：

主要设计思想：将保存车票信息的文件内容读入链表中。

其核心算法流程图如下：

输入火车信息

检查火车是否存在

否 是

是 火车信息保存于链表

是否继续

否

保存于文件

4 void QueryTrain( )函数：

其核心算法流程图如下：

输入1或2

1 2

输入火车号 输入目的地

返回

5 void UpdateInfo( ) 函数：

其核心算法流程图如下：

输入1或2

1 2

输入火车号 输入目的地

是否修改

是 否

输入新的火车信息

保存于文件

返回

四、**调试分析**

1. 在输入时会不小心多打空格进入，使得程序结果有差错。

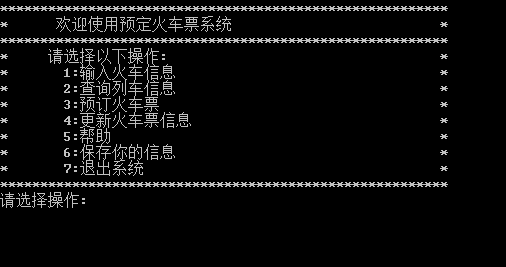
2. 初始时未加上#include <stdlib.h>和#include <string.h> 导致出错。

3. 编程过程中容易忽视C语言应有的规范，导致错误，后经调试，修正过来。

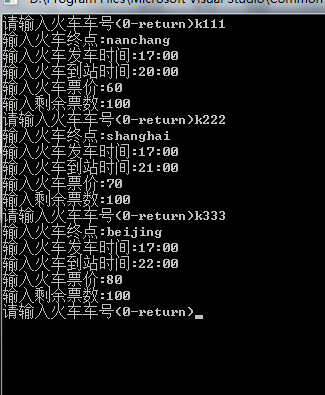
4. 这次编程内容需要自学，让我从中发现更多编程语言的乐趣，并且，我了解到应该以更严谨的态度去学习C语言。

1. **用户手册**
2. 程序编译环境为VC6.0，操作环境为DOS操作系统。
3. 本程序运行时输入后必须保存后，下次才能直接使用。
4. 进入菜单后，根据提示选择1-7按键，依据自己所需操作。
5. 必须按照输入时的目的地或火车号才能搜到用户需要的信息。
6. 进入程序，系统会显示主要选择功能菜单界面。在第一个选择界面里，选择用户自己需要的选项，输入相应数字，系统会谈跳出相应的界面，用户按相应提示操作即可。
7. **测试数据**

1.进入菜单

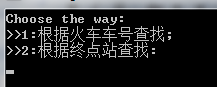


选择1，进入火车信息输入。

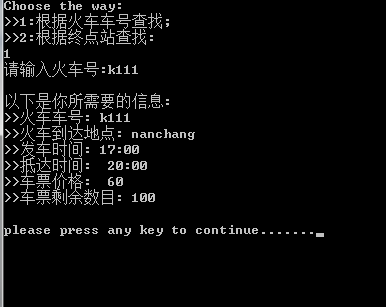


按0返回菜单。

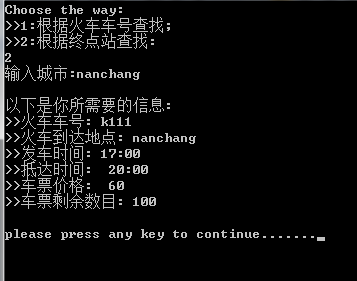
按2 查询火车信息。



输入1 再输入火车号k111



输入2 后输入 nanchang



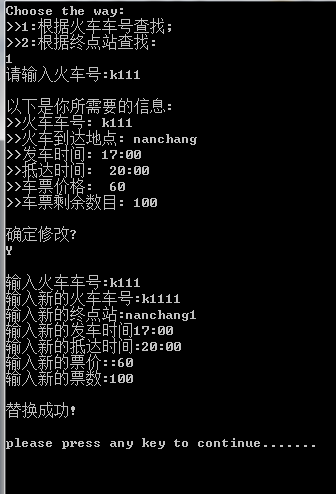
返回菜单后选择3预定火车票

输入 Beijing 输入1 和自己信息 123 123 票数为1



返回菜单输入4更新火车信息

更改k111 信息 输入Y更改 将k111改为k1111 nanchang改为nanchang1



选择5帮助选择火车



根据提示选择4预定改火车票

选择6保存你的信息后选择7可退出。

**七．附录：**

源程序：

#include <conio.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int shoudsave=0 ;

int count1=0,count2=0,mark=0,mark1=0 ;

/\*定义存储火车信息的结构体\*/

struct train

{

char num[10];/\*列车号\*/

char city[10];/\*目的城市\*/

char takeoffTime[10];/\*发车时间\*/

char receiveTime[10];/\*到达时间\*/

int price;/\*票价\*/

int bookNum ;/\*票数\*/

};

/\*订票人的信息\*/

struct man

{

char num[10];/\*ID\*/

char name[10];/\*姓名\*/

int bookNum ;/\*需求的票数\*/

};

/\*定义火车信息链表的结点结构\*/

typedef struct node

{

struct train data ;

struct node \* next ;

}Node,\*Link ;

/\*定义订票人链表的结点结构\*/

typedef struct people

{

struct man data ;

struct people\*next ;

}bookMan,\*bookManLink ;

/\* 初始界面\*/

void printInterface()

{

puts("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

puts("\* 欢迎使用预定火车票系统 \*");

puts("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

puts("\* 请选择以下操作: \*");

puts("\* 1:输入火车信息 \*");

puts("\* 2:查询列车信息 \*");

puts("\* 3:预订火车票 \*");

puts("\* 4:更新火车票信息 \*");

puts("\* 5:帮助 \*");

puts("\* 6:保存你的信息 \*");

puts("\* 7:退出系统 \*");

puts("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

/\*添加一个火车信息\*/

void InsertTraininfo(Link linkhead)

{

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

char num[10];

r = linkhead ;

s = linkhead->next ;

while(r->next!=NULL)

r=r->next ;

while(1)

{

printf("请输入火车车号(0-return)");

scanf("%s",num);

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

break ;

/\*判断是否已经存在\*/

while(s)

{

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

{

printf("火车 '%s'已经保存!\n",num);

return ;

}

s = s->next ;

}

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

strcpy(p->data.num,num);

printf("输入火车终点:");

scanf("%s",p->data.city);

printf("输入火车发车时间:");

scanf("%s",p->data.takeoffTime);

printf("输入火车到站时间:");

scanf("%s",&p->data.receiveTime);

printf("输入火车票价:");

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

printf("输入剩余票数:");

scanf("%d",&p->data.bookNum);

p->next=NULL ;

r->next=p ;

r=p ;

shoudsave = 1 ;

}

}

/\*打印火车票信息\*/

void printTrainInfo(struct node\*p)

{

puts("\n以下是你所需要的信息:");

printf(">>火车车号: %s\n",p->data.num);

printf(">>火车到达地点: %s\n",p->data.city);

printf(">>发车时间: %s\n>>抵达时间: %s\n",p->data.takeoffTime,p->data.receiveTime);

printf(">>车票价格: %d\n",p->data.price);

printf(">>车票剩余数目: %d\n",p->data.bookNum);

}

struct node \* Locate1(Link l,char findmess[],char numorcity[])

{

Node\*r ;

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

{

r=l->next ;

while(r)

{

if(strcmp(r->data.num,findmess)==0)

return r ;

r=r->next ;

}

}

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

{

r=l->next ;

while(r)

{

if(strcmp(r->data.city,findmess)==0)

return r ;

r=r->next ;

}

}

return 0 ;

}

/\*查询火车信息\*/

void QueryTrain(Link l)

{

Node \*p ;

int sel ;

char str1[5],str2[10];

if(!l->next)

{

printf("没有任何信息 !");

return ;

}

printf("Choose the way:\n>>1:根据火车车号查找;\n>>2:根据终点站查找:\n");

scanf("%d",&sel);

if(sel==1)

{

printf("请输入火车号:");

scanf("%s",str1);

p=Locate1(l,str1,"num");

if(p)

{

printTrainInfo(p);

}

else

{

mark1=1 ;

printf("\n没有找到!");

}

}

else if(sel==2)

{

printf("输入城市:");

scanf("%s",str2);

p=Locate1(l,str2,"city");

if(p)

{

printTrainInfo(p);

}

else

{

mark1=1 ;

printf("\n没有找到!");

}

}

}

/\*订票子模块\*/

void BookTicket(Link l,bookManLink k)

{

Node\*r[10],\*p ;

char ch,dem ;

bookMan\*v,\*h ;

int i=0,t=0 ;

char str[10],str1[10],str2[10];

v=k ;

while(v->next!=NULL)

v=v->next ;

printf("输入终点站: ");

scanf("%s",&str);

p=l->next ;

while(p!=NULL)

{

if(strcmp(p->data.city,str)==0)

{

r[i]=p ;

i++;

}

p=p->next ;

}

printf("\n\nthe number of record have %d\n",i);

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

printTrainInfo(r[t]);

if(i==0)

printf("\n\t\t\t对不起!没有找到该火车信息\n");

else

{

printf("\n是否预定?<1/0>\n");

scanf("%d",&ch);

if(ch == 1)

{

h=(bookMan\*)malloc(sizeof(bookMan));

printf("输入姓名: ");

scanf("%s",&str1);

strcpy(h->data.name,str1);

printf("输入身份证号: ");

scanf("%s",&str2);

strcpy(h->data.num,str2);

printf("输入预定数目: ");

scanf("%d",&dem);

h->data.bookNum=dem ;

h->next=NULL ;

v->next=h ;

v=h ;

printf("\n恭喜，您已经预定成功!");

getch();

shoudsave=1 ;

}

}

}

bookMan\*Locate2(bookManLink k,char findmess[])

{

bookMan\*r ;

r=k->next ;

while(r)

{

if(strcmp(r->data.num,findmess)==0)

{

mark=1 ;

return r ;

}

r=r->next ;

}

return 0 ;

}

/\*修改火车信息\*/

void UpdateInfo(Link l)

{

Node\*p ;

char findmess[20],ch ;

if(!l->next)

{

printf("\nthere isn't record for you to modify!\n");

return ;

}

else

{

QueryTrain(l);

if(mark1==0)

{

printf("\n确定修改?\n");

getchar();

scanf("%c",&ch);

if(ch=='y');

{

printf("\n输入火车车号:");

scanf("%s",findmess);

p=Locate1(l,findmess,"num");

if(p)

{

printf("输入新的火车车号:");

scanf("%s",&p->data.num);

printf("输入新的终点站:");

scanf("%s",&p->data.city);

printf("输入新的发车时间");

scanf("%s",&p->data.takeoffTime);

printf("输入新的抵达时间:");

scanf("%s",&p->data.receiveTime);

printf("输入新的票价::");

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

printf("输入新的票数:");

scanf("%d",&p->data.bookNum);

printf("\n替换成功!\n");

shoudsave=1 ;

}

else

printf("\t\t\t没有找到!");

}

}

else

mark1=0 ;

}

}

/\*系统给用户的提示信息\*/

void AdvicedTrains(Link l)

{

Node\*r ;

char str[10];

int mar=0 ;

r=l->next ;

printf("输入终点站: ");

scanf("%s",str);

while(r)

{

if(strcmp(r->data.city,str)==0&&r->data.bookNum<200)

{

mar=1 ;

printf("\n你可以选择以下火车!\n");

printf("\n\n请选择4预定该车票!\n");

printTrainInfo(r);

}

r=r->next ;

}

if(mar==0)

printf("\n\t\t\t对不起，现在无法预定!\n");

}

/\*保存火车信息\*/

void SaveTrainInfo(Link l)

{

FILE\*fp ;

Node\*p ;

int count=0,flag=1 ;

fp=fopen("c:\\train.txt","wb");

if(fp==NULL)

{

printf("文件无法打开!");

return ;

}

p=l->next ;

while(p)

{

if(fwrite(p,sizeof(Node),1,fp)==1)

{

p=p->next ;

count++;

}

else

{

flag=0 ;

break ;

}

}

if(flag)

{

printf("the number of the record which have been saved is %d\n",count); //1111

shoudsave=0 ;

}

fclose(fp);

}

/\*保存订票人的信息\*/

void SaveBookmanInfo(bookManLink k)

{

FILE\*fp ;

bookMan\*p ;

int count=0,flag=1 ;

fp=fopen("c:\\man.txt","wb");

if(fp==NULL)

{

printf("文件无法打开!");

return ;

}

p=k->next ;

while(p)

{

if(fwrite(p,sizeof(bookMan),1,fp)==1)

{

p=p->next ;

count++;

}

else

{

flag=0 ;

break ;

}

}

if(flag)

{

printf("the number of the record which have been saved is %d\n",count); //111111

shoudsave=0 ;

}

fclose(fp);

}

int main()

{

FILE\*fp1,\*fp2 ;

Node\*p,\*r ;

char ch1,ch2 ;

Link l ;

bookManLink k ;

bookMan\*t,\*h ;

int sel ;

l=(Node\*)malloc(sizeof(Node));

l->next=NULL ;

r=l ;

k=(bookMan\*)malloc(sizeof(bookMan));

k->next=NULL ;

h=k ;

fp1=fopen("c:\\train.txt","ab+");

if((fp1==NULL))

{

printf("无法打开文件!");

return 0 ;

}

while(!feof(fp1))

{

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

if(fread(p,sizeof(Node),1,fp1)==1)

{

p->next=NULL ;

r->next=p ;

r=p ;

count1++;

}

}

fclose(fp1);

fp2=fopen("c:\\man.txt","ab+");

if((fp2==NULL))

{

printf("无法打开文件!");

return 0 ;

}

while(!feof(fp2))

{

t=(bookMan\*)malloc(sizeof(bookMan));

if(fread(t,sizeof(bookMan),1,fp2)==1)

{

t->next=NULL ;

h->next=t ;

h=t ;

count2++;

}

}

fclose(fp2);

while(1)

{

system("cls");

printInterface();

printf("请选择操作: ");

scanf("%d",&sel);

system("cls");

if(sel==8)

{

if(shoudsave==1)

{

getchar();

printf("\n文件已经改变，是否保存？(y/n)?\n");

scanf("%c",&ch1);

if(ch1=='y'||ch1=='Y')

{

SaveBookmanInfo(k);

SaveTrainInfo(l);

}

}

printf("\n谢谢\n");

break ;

}

switch(sel)

{

case 1 :

InsertTraininfo(l);break ;

case 2 :

QueryTrain(l);break ;

case 3 :

BookTicket(l,k);break ;

case 4 :

UpdateInfo(l);break ;

case 5 :

AdvicedTrains(l);break ;

case 6 :

SaveTrainInfo(l);SaveBookmanInfo(k);break ;

case 7 :

return 0;

}

printf("\nplease press any key to continue.......");

getch();

}

return 0;

}