# 航空订票系统

## 1.需求分析

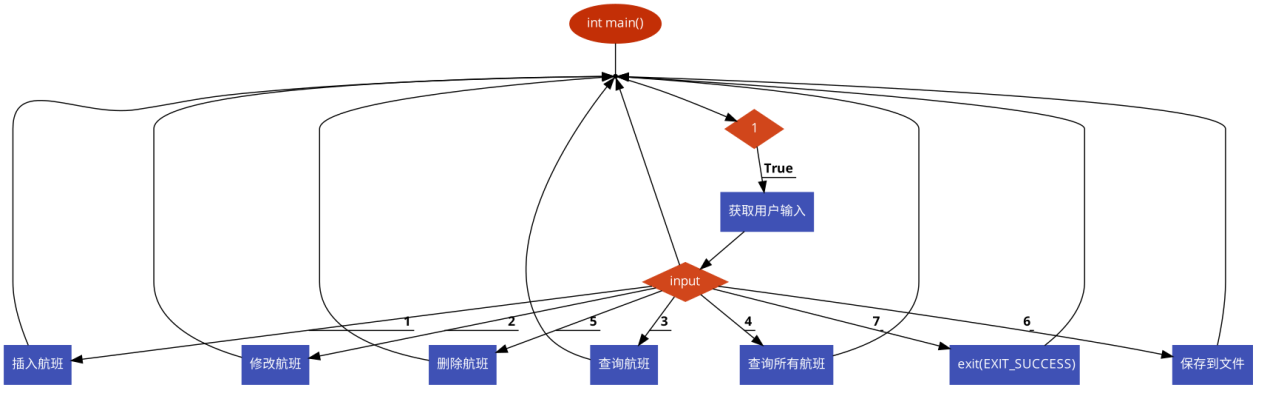
按照题目要求，使用单链表single linked list实现对航班信息数据的增加，删除，修改和查询等工作。使用stdio提供的printf和puts函数进行输出，通过fopen函数进行对数据的保存和数据的操作。

## 系统框架思维导图

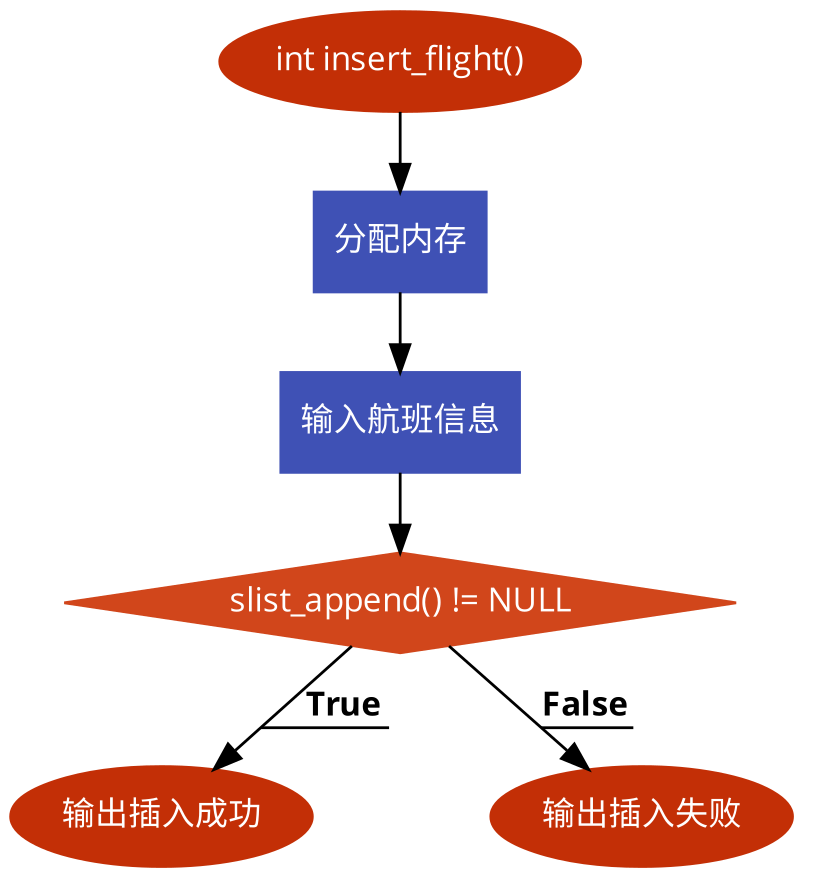


## 程序流程图

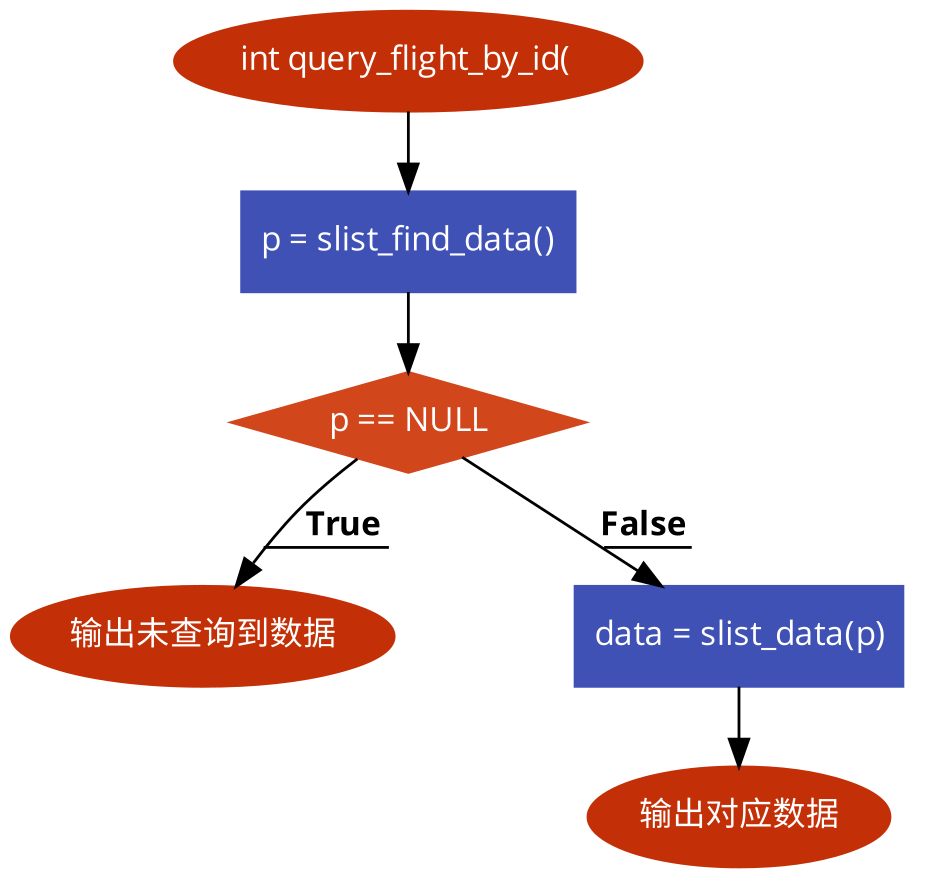
主函数代码:



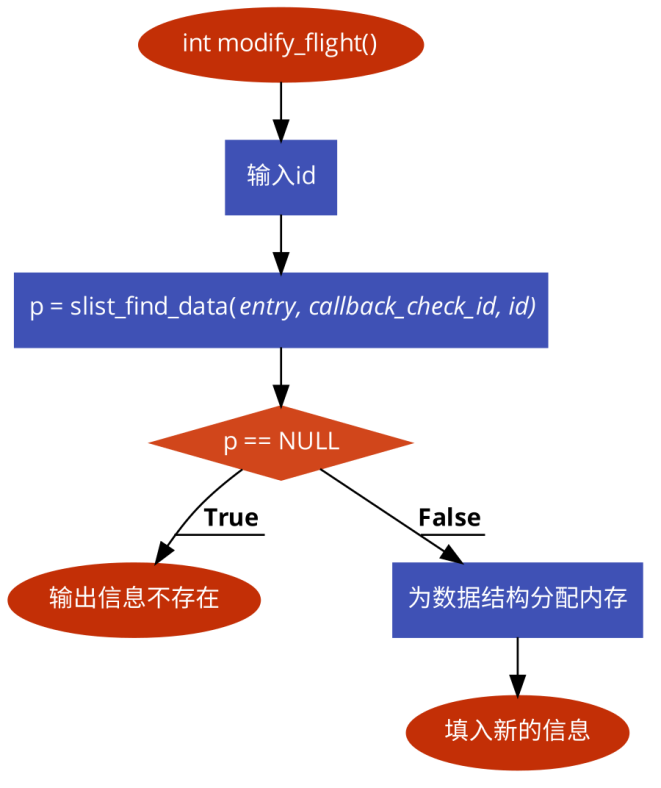
插入航班流程图如下：



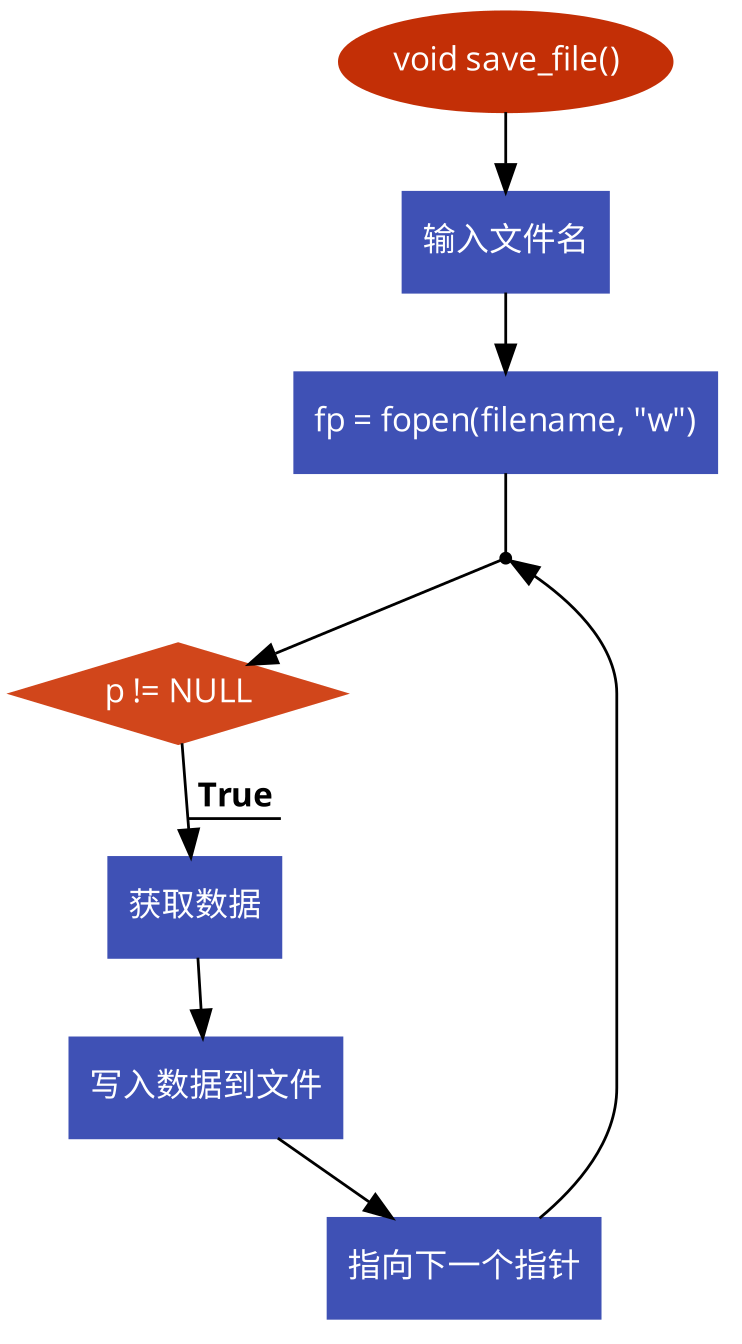
查询航班流程图:



修改航班:



保存到文件:



## 重要代码分析

主程序:

本函数，主要使用switch case驱动，让用户输入选项，并跳转到相应函数，代码如下，

while (1) {

welcome();

puts("");

puts("输入你的选择:");

scanf("%d", &input);

switch (input) {

case 1:

insert\_flight(&flightList);

break;

case 2:

modify\_flight(&flightList);

break;

case 3:

query\_flight(&flightList);

break;

case 4:

show\_flight\_table(&flightList);

break;

case 5:

delete\_flight(&flightList);

break;

case 6:

save\_file(&flightList);

break;

case 7:

exit(EXIT\_SUCCESS);

default:

puts("-------------------");

puts("输入错误");

puts("-------------------");

break;

}

插入航班:

使用malloc分配内存，使用memset进行数据清零，防止随机数据的产生:

flight\_info = malloc(sizeof (struct \_Flight));

memset(flight\_info, 0, sizeof(struct \_Flight));

if (flight\_info == NULL) {

perror("malloc");

exit(EXIT\_FAILURE);

}

使用slist\_append函数，添加到链表

slist\_append(list, (SListValue \*)flight\_info)

删除航班:

先使用slist\_find\_data，查询数据是否存在，如果存在则使用slist\_remove\_entry删除，

p = slist\_find\_data(\*entry, callback\_check\_id, (SListValue \*)&id);

if (p == NULL) {

puts("-------------------");

puts("航班不存在");

puts("-------------------");

return -1;

} else {

value = slist\_data(p);

flight\_info = (struct \_Flight \*) value;

if (slist\_remove\_entry(entry, p) == 0) {

puts("-------------------");

puts("删除数据失败");

puts("-------------------");

return -1;

} else {

puts("-------------------");

puts("删除数据成功");

puts("-------------------");

return 1;

}

}

查询所有数据:

使用slist\_find\_data查询数据，如果存在则进行修改，

p = slist\_find\_data(\*entry, callback\_check\_id, (SListValue \*)&id);

if (p == NULL) {

puts("-------------------");

puts("未查询到数据");

puts("-------------------");

return -1;

} else {

data = slist\_data(p);

puts("----------------------");

printf("查询结果如下\n");

// 这里需要强制转换

printf("id:%d\n", ((struct \_Flight \*)data)->id);

printf("开始时间:%s\n", ((struct \_Flight \*)data)->startTime);

printf("到达时间:%s\n", ((struct \_Flight \*)data)->stopTime);

printf("机票价格:%lf\n", ((struct \_Flight \*)data)->price);

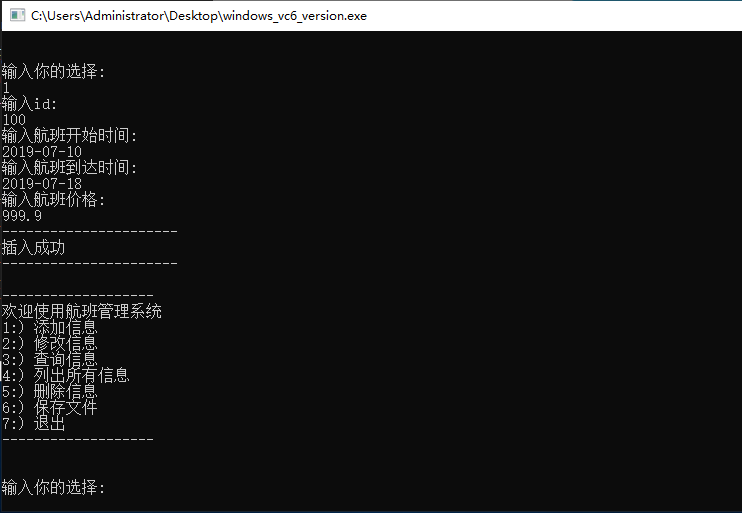
puts("----------------------");

}

其他的代码与上面类似。

## 运行结果

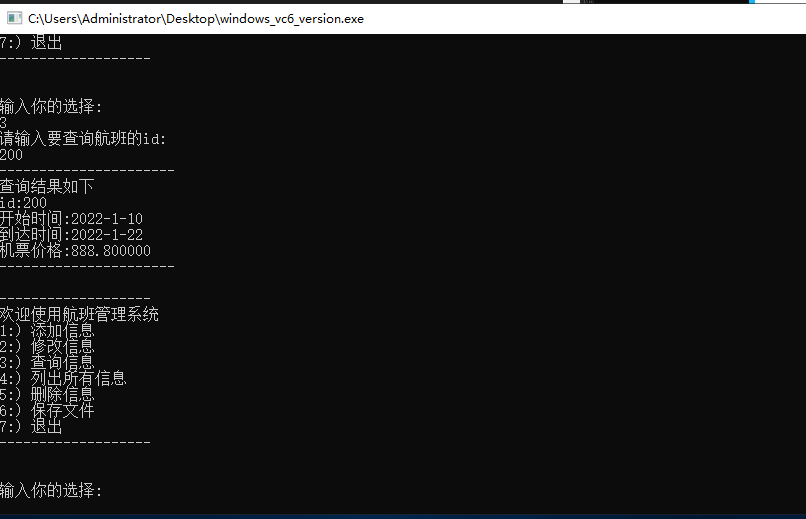
插入航班:



修改航班:



查询航班:



保存到文件:

