

# 数据结构\_基于链表的通讯录

---

顺序表的源代码需要略作修改，如下 将数据类型改为通讯录的结构体。注释掉打印，查找的函数。

## SList.h

---

```
#define _CRT_SECURE_NO_WARNINGS 1

#include<stdio.h>
#include<stdlib.h>
#include<assert.h>
#include"Contact.h"

typedef PeoInfo SList_Datatype;

typedef struct SList_Node
{
    SList_Datatype Data;
    struct SList_Node* Next;
}STL_Node;

//void SLTPrint(STL_Node* phead);

//尾插
void SLTPushBack(STL_Node** pphead, SList_Datatype x);
//头插
void SLTPushFront(STL_Node** pphead, SList_Datatype x);
//尾删
void SLTPopBack(STL_Node** pphead);
//头删
void SLTPopFront(STL_Node** pphead);

//查找
//STL_Node* SLTFind(STL_Node* phead, SList_Datatype x);

//在指定位置之前插入数据
void SLTInsert(STL_Node** pphead, STL_Node* pos, SList_Datatype x);
//在指定位置之后插入数据
void SLTInsertAfter(STL_Node* pos, SList_Datatype x);

//删除pos节点
void SLTErase(STL_Node** pphead, STL_Node* pos);
//删除pos之后的节点
void SLTEraseAfter(STL_Node* pos);

//销毁链表
void SListDesTroy(STL_Node** pphead);
```

# Contact.h

---

新建一个结构体，存储联系人的数据。将链表重新给一个名字contact。

实现以下几个接口：

```
#pragma once

#define NAME_MAX 20
//前置声明
typedef struct SList_Node contact;
//用户数据
typedef struct PersonInfo
{
    char name[NAME_MAX];
    char tel[11];
}PeoInfo;

//初始化通讯录
void InitContact(contact** con);
//添加通讯录数据
void AddContact(contact** con);
//删除通讯录数据
void DelContact(contact** con);
//展示通讯录数据
void ShowContact(contact* con);
//查找通讯录数据
contact* FindContact(contact* con, char* find);
//修改通讯录数据
void ModifyContact(contact** con);
//销毁通讯录数据
void DestroyContact(contact** con);
```

# Contact.c

---

因为我写过一遍通讯录（基于顺序表）所以再基于链表写一个的难度不是很大。

这里我要特别提醒自己和在看的各位：**if(xxx == xxx) 中间一定记得用双等号，不要手滑搞成了 '='**（因为本人因为这个错误找了半小时bug，知道是这个原因的时候直接碎掉）

```
#define _CRT_SECURE_NO_WARNINGS 1
#include"Contact.h"
#include"SList.h"
#include<stdio.h>
```

```
#include<string.h>
//初始化通讯录
void InitContact(contact** con)
{
    contact* phead = NULL;
    *con = phead;
}
//添加通讯录数据
void AddContact(contact** con)
{
    PeoInfo newperson;
    printf("姓名: ");
    scanf("%s", newperson.name);
    printf("电话: ");
    scanf("%s", newperson.tel);
    SLTPushBack(con, newperson);
}
//删除通讯录数据
void DelContact(contact** con)
{
    char del[NAME_MAX];
    printf("要删除的联系人的姓名: ");
    scanf("%s", del);

    contact* find = FindContact(*con, del);

    if (find == NULL)
    {
        printf("要删除的联系人不存在");
        return;
    }

    SLTErase(con, find);
}
//展示通讯录数据
void ShowContact(contact* con)
{
    assert(con);
    while (con)
    {
        printf("姓名: %s\n", con->Data.name);
        printf("电话: %s\n", con->Data.tel);
        con = con->Next;
    }
}
//查找通讯录数据
contact* FindContact(contact* con, char* find)
{
    contact* pcur = con;
    while (pcur)
    {
        if (strcmp(find, pcur->Data.name) == 0)
        {
            return pcur;
        }
    }
}
```

```
        }
        pcur = pcur->Next;
    }
    return NULL;
}
//修改通讯录数据
void ModifyContact(contact** con)
{
    printf("请输入要修改联系人的名字: ");
    char mod[NAME_MAX];
    scanf("%s", mod);
    contact* find = FindContact(*con, mod);
    if (find == NULL)
    {
        printf("要修改的联系人不存在! ");
        return;
    }
    PeoInfo newperson;
    printf("请输入修改后的名字、电话: ");
    scanf("%s%s", newperson.name, newperson.tel);
    SLTInsertAfter(find, newperson);
    SLTErase(con, find);
}
//销毁通讯录数据
void DestroyContact(contact** con)
{
    SListDesTroy(con);
}
```

本博客旨在记录学习过程，以后忘了随时来看。