通讯录设计实验报告

实验日期：**2021.11.6** 班级：计科**201**

学号**1**：**20402070113**姓名**1**：瞿子椋

学号**2**：**20402080240**姓名**2**：王夏禹

学号**3**：**20401010121**姓名**3**：徐志伟

# 实验目的

用基本运算实现对线性表的具体运算

掌握文件操作的应用

提高对链接存储结构的数据结构的理解，逐步培养解决实际问题的编程能力

# 实验环境

一台使用**visualc++6.0/CFree**的计算机

# 实验内容

Address list design

Design a classmate's address list, requested as follows:

* Each student in the address list contains the following information: student id、name、telephone number. If you need more fields, please add them yourself.
* The program has a main menu containing the following functions:

1. Add a record: Add a student record from the input.
2. Delete a record: Delete a student record according to the student id from the input.
3. Output all records: Display all the records in the address list.
4. Search by name: Input the student name and then output the whole information of the student.
5. Save records: Save all the records in the address list to a certain file.
6. Clear records: Delete all the records in the address list and then delete the file.
7. Quit

# 重要的数据结构

//学生信息节点

typedef struct node

{

int sid;

char name[10];

int number;

node \* next;

}\*pNode;

//添加学生信息

void addSinfo(pNode& L)

{

pNode temp;

temp=(pNode)malloc(sizeof(struct node));

if(temp==NULL)

{

cout<<"空间不足，升级内存。"<<endl;

}

temp->next=NULL;

char name1[10];

int sid1;

int number1;

cout<<"请输入学生姓名："<<endl;

cin>>name1;

strcpy(temp->name, name1);//用=会报错

cout<<"请输入学生ID："<<endl;

cin>>sid1;

temp->sid=sid1;

cout<<"请输入学生电话号码："<<endl;

cin>>number1;

temp->number=number1;

pNode p;//创造一个工作指针

p=L;

while(p->next!=NULL)

{

p=p->next;

}

p->next=temp;

}

//删除学生的信息

void deleteinfo(pNode& L)

{

pNode temp;

cout<<"请输入要删除学生的账号："<<endl;

int id;

cin>>id;

pNode p=L->next;

if(p->sid==id)

{

temp=p;

L->next=p->next;

free(p);

}

else

{

while(p->next!=NULL)

{

if(p->next->sid==id)

{

temp=p->next;

L->next=p->next->next;

free(temp);

break;

}

p=p->next;

}

}

}

//输出所有学生的信息

void outputall(pNode& L)

{

pNode p=L->next;

while(p!=NULL)

{

cout<<"姓名："<<p->name<<endl;

cout<<"学号："<<p->sid<<endl;

cout<<"电话："<<p->number<<endl;

p=p->next;

}

}

//根据姓名查找

void findstudent(pNode& L)

{

pNode p=L->next;

cout<<"请输入你想找的学生姓名："<<endl;

char name2[10];

cin>>name2;

while(p!=NULL&&strcmp(p->name, name2))

{

p=p->next;

}

if(p==NULL)

{

cout<<"查无此人"<<endl;

}

else

{

cout<<"学生的姓名："<<p->name<<endl;

cout<<"学生的学号："<<p->sid<<endl;

cout<<"学生的电话："<<p->number<<endl;

}

}

//保存学生的信息

void saveinfo(pNode& L)

{

fstream ofs;

ofs.open("studentinfo.txt",ios::out);

pNode p=L->next;

cout<<"保存中..."<<endl;

while(p!=NULL)

{

ofs<<p->name<<" "<<p->sid<<" "<<p->number<<" "<<endl;

p=p->next;

cout<<"\*"<<endl;

}

ofs.close();

cout<<"保存成功！"<<endl;

}

//清除数据

void clearinfo(pNode& L)

{

pNode temp;

pNode p=L->next;

L->next=NULL;//不能用free(L);free只是让内存释放，但他指向的东西并非NULL，而是乱码

while(p!=NULL)

{

temp=p->next;

p->next=NULL;//同理

p=temp;

}

cout<<"清除成功！"<<endl;

}

# 实现分析

通过线性表创建一个个存储学生信息的结点，在通过单链表的特性，将一个个结点链接起来，实现了一个存储学生信息的数据库。可以写入和读取txt文件中的数据。

# 调试问题分析

在一开始实验时进行了许多次运行代码，但是总会有很多次程序出错的情况。当调错多次之后，终于发现了还是内存泄漏的原因。

详细情况如下：

void clearinfo(pNode& L)

{

pNode temp;

pNode p=L->next;

L->next=NULL;//不能用free(L);free只是让内存释放，但他指向的东西并非NULL，而是乱码

while(p!=NULL)

{

temp=p->next;

p->next=NULL;//同理

p=temp;

}

cout<<"清除成功！"<<endl;

}

# 总结

首先熟悉了link list的知识。将读取文件的知识与链表结合在一起使用。对于内存空间的分配进一步了解，区别了free()和Array->=NULL；的区别。

1. 工作划分

|  |  |  |
| --- | --- | --- |
|  | 分组 |  |
| 成员名称 | 完成的工作 | 完成情况 |
| 瞿子椋 | 文件的读写部分 | 已完成 |
| 王夏禹 | **程序的完成和调试** | 已完成 |
| 徐志伟 | 完成实验报告 | 已完成 |