通信与信息工程学院

《数据结构》大作业

|  |  |
| --- | --- |
| 班 级： | 电信1702 |
| 姓 名： | 秦伟钦 |
| 学 号： | 17407110815 |

通信与信息工程学院

二〇一九年

1. 设计要求

本次实验是为了完成图书管图书信息管理系统，并将图书信息以顺序表的形 式写入txt文件里，并在下次打开此程序时，能继续对图书信息进行增删查改等功能。

1. 设计思路

首先要完成这样的程序的设计，我们得先确定我们要用什么样的存储结构，这里我使用了顺序表的存储结构，并定义了一个图书信息的结构体，将这个结构体放入确定好的顺序表内，之后我就开始设计关于这个顺序表信息的增删查改的功能设计。再将这些功能子函数在main函数中调用，实现我们需要的功能，做了之后，我差不多完成了程序的所有的功能，然后我们要将信息存入文件之中，因为如果不将信息存储到文件中，一旦我们关闭了这个程序，程序的内存就会释放，我们的图书信息就不能得到保存。所以我们要写读写文件的子函数以完成这项功能。

1. 源程序

work.h：

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<assert.h>

#include<errno.h>

#define PEO\_MAX 2   //自定义刚开始的容量

#define NAME\_MAX 20 //自定义为名字分配空间的大小

enum num //枚举常量

{

    esc,

    add,

    dele,

    search,

    show,

    modifi,

    sort,

};

typedef struct library

{

    char name[NAME\_MAX];//图书

  float fee;//费用

}library;

typedef struct contact

{

    library \*book;//维护动态开辟的内存；

    int count;//当前图书的数量

    int sz;//当前空间的容量

}contact;

void init\_contact(contact\* pcon);//初始化图书信息列表

void add\_contact(contact\* pcon);//添加图书信息

void show\_contact(const contact\* pcon);//显示图书信息

void dele\_contact(contact\* pcon);//删除图书信息

void serch\_contact(const contact\* pcon);//查询图书信息

void modifi\_contact(contact\* pcon);//修改图书信息

void sort\_contact(contact\* pcon);//排序图书信息

int CheckCapacity(contact\* pcon);//动态内存扩容（检查是否需要扩容）

void DestroyContact(contact\* pcon);//释放动态内存

void save\_contact(contact\* pcon);//把数据保存到文件

void read\_contact(contact\* pcon);//读取数据到程序

work.c：

#include"work.h"

void init\_contact(contact\* pcon)//初始化列表

{

    assert(pcon != NULL);

    pcon->count = 0;//初始化个数

    pcon->sz = PEO\_MAX;

    pcon->book = (library\*)malloc(PEO\_MAX\*sizeof(library));//为列表创建动态内存

    if (pcon->book == NULL)

    {

        printf("%s\n", strerror(errno));//打印错误信息

        return;

    }

    memset(pcon->book, '0', PEO\_MAX\*sizeof(library));

    read\_contact(pcon);

    return;

}

void add\_contact(contact\* pcon)//添加图书信息

{

    assert(pcon != NULL);

    if (1 == CheckCapacity(pcon))

    {

        printf("请输入图书的名字：  ");

        scanf("%s", pcon->book[pcon->count].name);

        printf("请输入图书的费用：  ");

        scanf("%f", &(pcon->book[pcon->count].fee));

        pcon->count++;

        return;

    }

}

void show\_contact(const contact \*pcon)//查看图书信息

{

    assert(pcon != NULL);

    if (pcon->count == 0)

    {

        printf("图书信息列表为空！\n");

        return;

    }

    int i = 0;

    printf("%-10s\t%-10s\t\n", "图书姓名", "费用");

    for (i = 0; i < pcon->count; i++)

    {

        printf("%-10s\t%-10f\t\n", pcon->book[i].name,pcon->book[i].fee);

    }

    return;

}

void dele\_contact(contact \*pcon)//删除图书信息

{

    assert(pcon != NULL);

    int i = 0;

    int j = 0;

    char name[20];

    if (pcon->count == 0)

    {

        printf("图书信息列表为空，无法删除！\n");

        return;

    }

    printf("请输入要删除图书的名字：");

    scanf("%s", name);

    for (i = 0; i < pcon->count; i++)

    {

        if (strcmp(name, pcon->book[i].name) == 0)

        {

            for (j = i; j < pcon->count; j++)

            {

                pcon->book[j] = pcon->book[j + 1];

            }

            pcon->count--;

            printf("删除成功！\n");

            return;

        }

    }

    printf("图书信息列表中没有此书！\n");

    return;

}

void serch\_contact(const contact\* pcon)//查询图书信息

{

    int input = 0;

    printf("按图书姓名还是按图书费用查询？（图书姓名请输入 1，图书费用请输入 2）\n");

    scanf("%d", &input);

    switch (input){

        case 1:

        {

            int flag=0;

            char name[20];

            scanf("%s", name);

    for (int i = 0; i < pcon->count; i++)

    {

        if (strcmp(name, pcon->book[i].name) == 0){

             printf("%-10s\t%-10f\t\n", pcon->book[i].name,pcon->book[i].fee);

             flag=1;

        }

    }

     if(flag==1)

        break;

    printf("没有这本书！\n");

     break;

        }

        case 2:

        {

            int flag=0;

            float fee;

            scanf("%f",&fee);

            for (int i = 0; i < pcon->count; i++){

                if(fee==pcon->book[i].fee)

               { printf("%-10s\t%-10f\t\n", pcon->book[i].name,pcon->book[i].fee);

               flag=1;}

        }

         if(flag==1)

        break;

       printf("没有这本书！\n");

        break;

        }

    }

}

void modifi\_contact(contact\* pcon)//修改图书信息

{

    assert(pcon != NULL);

    int i = 0;

    char name[20];

    if (pcon->count == 0)

    {

        printf("图书列表为空，无法修改\n");

        return;

    }

    printf("请输入要修改图书的名字：");

    scanf("%s", name);

    for (i = 0; i < pcon->count; i++)

    {

        if (strcmp(name, pcon->book[i].name) == 0)

        {

            printf("请输入修改后图书的姓名：  ");

            scanf("%s", pcon->book[i].name);

            printf("请输入修改后图书的费用：  ");

            scanf("%f", &(pcon->book[i].fee));

            printf("修改成功！\n");

            return;

        }

    }

    printf("图书列表中没有此书！\n");

    return;

}

void sort\_contact(contact\* pcon)//给图书排序

{

    int input = 0;

    printf("按照姓名排序请输入 1，按照费用排序请输入 2\n");

    scanf("%d", &input);

    switch (input)

    {

        case 1:

            assert(pcon != NULL);

            int i = 0;

            int j = 0;

            int flag = 0;

            library tmp;

            if (pcon->count == 0)

            {

                printf("图书列表为空，无法修改\n");

                return;

            }

            for (i = 0; i < pcon->count; i++)//确定趟数

            {

                flag = 1;

                for (j = 0; j < (pcon->count - i - 1); j++)

                {

                    if (strcmp(pcon->book[j].name, pcon->book[j + 1].name)>0)

                    {

                        tmp = pcon->book[j];

                        pcon->book[j] = pcon->book[j + 1];

                        pcon->book[j + 1] = tmp;

                        flag = 0;

                    }

                }

                if (flag == 1)

                {

                    printf("排序成功！\n");

                    return;

                }

            }

        case 2:

            assert(pcon != NULL);

            if (pcon->count == 0)

            {

                printf("图书列表为空，无法修改\n");

                return;

            }

            i = 0;

            j = 0;

            for (i = 0; i < pcon->count; i++)//确定趟数

            {

                flag = 1;

                for (j = 0; j < (pcon->count - i - 1); j++)

                {

                    if (pcon->book[j].fee>pcon->book[j+1].fee)

                    {

                        tmp = pcon->book[j];

                        pcon->book[j] = pcon->book[j + 1];

                        pcon->book[j + 1] = tmp;

                        flag = 0;

                    }

                }

                if (flag == 1)

                {

                    printf("排序成功！\n");

                    return;

                }

            }

        default:

            printf("选择错误！请重新选择！\n");

            return;

    }

}

int CheckCapacity(contact\* pcon)//检验是否需要扩容函数

{

    assert(pcon != NULL);

    if (pcon->sz == pcon->count)

    {

        library\* tmp = (library\*)realloc(pcon->book, (pcon->sz + 2)\*sizeof(library));

        if (tmp != NULL)

        {

            pcon->book = tmp;

            pcon->sz += 2;

            //printf("扩容成功！\n");//扩容成功，实际运行时不用打印；

            return 1;

        }

        else

        {

            printf("%s\n", strerror(errno));//打印错误信息

            return 0;

        }

    }

    else

    {

        return 1;

    }

}

void DestroyContact(contact\* pcon)//释放动态内存

{

    assert(pcon != NULL);

    free(pcon->book);

    pcon->book = NULL;

    pcon->sz = 0;

    pcon->count = 0;

}

void save\_contact(contact\* pcon)//把数据保存到文件

{

    assert(pcon != NULL);

    int i = 0;

    FILE\* pfin = fopen("name.txt", "wb");

    if (pfin == NULL)

    {

        perror("open file for write:");//打开文件失败后打印错误信息

        return;

    }

    for (i = 0; i < pcon->count; i++)

    {

        fwrite(pcon->book + i, sizeof(library), 1, pfin);

    }

    fclose(pfin);//关闭文件

    pfin = NULL;

    printf("保存成功\n");

    return;

}

void read\_contact(contact\* pcon)//读取文件中的数据到程序

{

    assert(pcon != NULL);

    library tmp = { 0 };

    FILE\* pfout = fopen("name.txt", "rb");

    if (pfout == NULL)

    {

        perror("open file for read:");

        return;

    }

    while (fread(&tmp, sizeof(library), 1, pfout))

    {

        CheckCapacity(pcon);//检查是否需要扩容

        pcon->book[pcon->count] = tmp;

        pcon->count++;

    }

    fclose(pfout);

    pfout = NULL;

    //printf("加载数据完成!\n");//加载数据完成，实际运行时不用展示；

    return;

}

test.c：#include"work.h"

void menu()

{

    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  WELCOME  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

    printf("\*\*\*\*\*\*\*   1.add            2.dele    \*\*\*\*\*\*\*\*\n");//添加图书信息   删除图书信息

    printf("\*\*\*\*\*\*\*   3.search         4.show    \*\*\*\*\*\*\*\*\n");//查询图书信息  查看图书信息

    printf("\*\*\*\*\*\*\*   5.modifi         6.sort    \*\*\*\*\*\*\*\*\n");//修改图书信息   排序图书信息

    printf("\*\*\*\*\*\*\*           0.esc              \*\*\*\*\*\*\*\*\n");//  退出程序

    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

void test()

{

    contact my\_contact = { 0 };

    init\_contact(&my\_contact);//初始化图书信息；

    int input = 0;

    do

    {

        menu();

        printf("请选择—>");

        scanf("%d", &input);

        switch (input)

        {

            case add:

                add\_contact(&my\_contact);//添加图书信息

                break;

            case dele:

                dele\_contact(&my\_contact);//删除图书信息

                break;

            case search:

                serch\_contact(&my\_contact);//查询图书信息

                break;

            case show:

                show\_contact(&my\_contact);//查看图书信息

                break;

            case modifi:

                modifi\_contact(&my\_contact);//修改图书信息

                break;

            case sort:

                sort\_contact(&my\_contact);//排序图书信息

                break;

            case esc:

                save\_contact(&my\_contact);//把数据保存到文件中

                DestroyContact(&my\_contact);//释放动态内存

                printf("退出程序\n");

                break;

            default:

                printf("选择错误！请重新选择！\n");

        }

    } while (input);

}

int main()

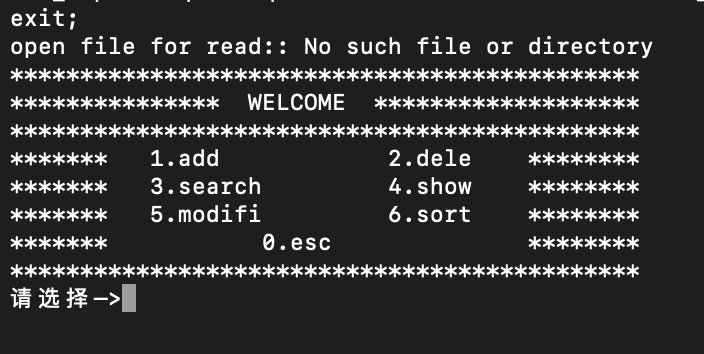
{

    test();

    return 0;

}

1. 运行结果及分析



进入程序，我们会进入到这样的界面

1. 添加图书的信息
2. 删除图书的信息
3. 查找图书
4. 展示如今图书库里的书
5. 修改图书的信息
6. 对图书的顺序进行排序
7. 退出这个程序，并将这个步骤中，我将保存信息至文件中

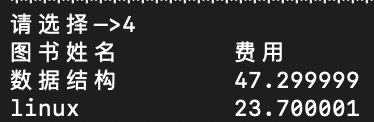
我进行第一步添加图书信息后，展示下我存储的信息



接下来删除信号处理这本书的信息

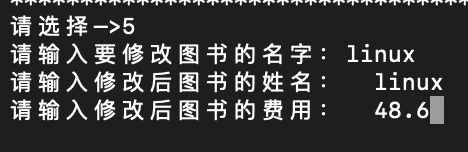


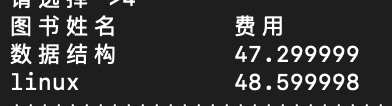
之后的图书信息为



成功删除

我们将linux的价格进行修改

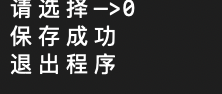




这是将书籍进行排序，通过名字的字符串比较或者费用的大小比较，由低到高排序，排序方法，我使用的是冒泡排序



退出这个程序时，会自动调用保存文件的函数



1. 总结和体会

通过这次的程序设计我更加了解了顺序表结构的建立过程，并在设计之中，最开始我是为顺序表中的book结构体申请的固定的空间，发现并不方便，之后我在查阅资料后，写了一个check内存的子函数，在我们进行add图书信息时，进行调用，若内存过半，则继续开辟空间，成为动态的空间开辟，当然我还是觉得，通过链表指针开辟空间的方式更加方便一些。在本次实验中，我觉得最值得我注意到是文件内容的读与写，因为以前接触的时候，没太在意，这次为了保存信息，在CSDN上查阅了关于C++读写文件函数的资料，这次实验程序设计用处很大，给我提供了实践代码的机会，练习，并提升自己代码的水平，并将这学期前段时间所学的数据结构内容得到细化，通过实践后，我更加容易理解。