Git账号：hallo128

图书管理系统笔记

图书管理系统的C语言程序：

https://github.com/hallo128/C/tree/master/BookDemo2

图书系统需求分析

- **操作用户：**图书系统管理员
- **功能**：实现对图书的管理

- 登录、退出系统
- 针对图书操作：增加、删除、查找、借出、归还、查看图书系统的所有图书

- **设计技巧**（事先规定）：

- 借阅状态与借出时间的关系：borrowTime=0——借阅状态：可借；borrowTime非0——借阅状态：不可借。（只在借出的时候，记录借出时间。一旦有借出时间记录，说明该本书的借阅状态为不可借）也就是，借出与归还时，只改变“借出时间”这一个属性。
- 自动编号id的设计要求：id一定是按从小到大，最后的id一定是最大的（不足：应该从所有图书中判断出最大id，以防最后不是最大）。id一旦确定再也不能更改。
- 最核心代码是模型层，数据与文件操作部分（FileOpt）。视图层负责界面的输出。业务层，就是模型层和视图层的整合，来更好的实现一个业务功能。
- 文件读写操作：

```
fread(保存结果的指针p1, 该数据类型的大小size, 数据个数n, 文件指针file);

//从文件中读操作：从file中，按size的大小读取n个，存放到p1中

fwrite(保存结果的指针p2, 该数据类型的大小, 数据个数, 文件指针file);

//写入文件的操作：从p1中，按size的大小读取n个，存放到file中
```

- **实现的界面**：

```
欢迎使用Extreme Acadamy图书管理系统
1、登录系统    2、退出
请选择：1
---------------
1、增加
2、删除
3、查找
4、借出
5、归还
6、查看图书系统的所有图书
---------------
欢迎进入图书管理系统!
请选择您想要进行的操作(选择1-6以外的任意键，可以退出系统)：6

所有图书信息如下：
-------------------------------------------------------------
内部编号    图书名称    出版号        借阅状态    图书单价    借出时间
-------------------------------------------------------------
1          狼图腾       456-897      可借        78.30     ---
2          三重门       564-897      可借        67.30     ---
3          寻味          560-098      可借        23.80     ---
4          老人与海     786-009      可借        45.23     ---
-------------------------------------------------------------
```

```
---------------
欢迎进入图书管理系统!
请选择您想要进行的操作(选择1-6以外的任意键，可以退出系统)：4
################(借出)
-------------------------可借状态的所有图书----------------------
内部编号    图书名称    出版号        借阅状态    图书单价    借出时间
-------------------------------------------------------------
1          狼图腾       456-897      可借        78.30     ---
2          三重门       564-897      可借        67.30     ---
3          寻味          560-098      可借        23.80     ---
4          老人与海     786-009      可借        45.23     ---
-------------------------------------------------------------
请输入要借出的图书编号：2
成功借出!
---------------
1、增加
2、删除
3、查找
4、借出
5、归还
6、查看图书系统的所有图书
---------------
欢迎进入图书管理系统!
请选择您想要进行的操作(选择1-6以外的任意键，可以退出系统)：3
-------------(查找)
请输入要查找的图书编号：2
-------------------------------------------------------------
内部编号    图书名称    出版号        借阅状态    图书单价    借出时间
-------------------------------------------------------------
2          三重门       564-897      不可借    67.30     2017-7-13  0: 7:29
-------------------------------------------------------------
```

特别地：

（1）视频代码问题更改：

- printAllBooks()不能放在Book.h中，应放在FileOpt.h中。
- memcpy用法（需要调用String.h头文件）：memcpy(tempBook, bookArray, bookCount*sizeof(Book))
- DeleteBookById函数：针对第一个位置、中间位置、最后位置，有不同的写法判别。
- UpdateBook函数：在读入临时变量tempBook 时，如果出现每次都只是读文件的第一个，那么应在前面再添加一行代码（保证每次读入tempBook时，都能从下个位置继续）fseek(file, i*sizeof(Book), SEEK_SET);
- 添加视图层的PrintBookStatus函数：根据借阅状态，打印数组中相应的图书数组（希望借阅时可以直接显示出当前可以操作的图书列表）

（2）我认为的难点部分：

- “FileOpt.c”中的GetAllBooks函数、DeleteBookById函数（随后文档会详细说明）

（3）我目前仍存在的疑问（如有小伙伴进行解答，将不甚感激）：

- GetAllBooks函数，前期设计时，可以用2种方式进行调用；但设计到后面，只能成功运行一种方法，另一种却不行。不知道是哪里的问题。

（可以运行的方式：方式二。不能运行：方式一）

```
int bookCount = -1;//图书总数
Book * bookArray = NULL;

//方式二：
bookArray = GetAllBooks(NULL, &bookCount);
//方式一：
//GetAllBooks(bookArray, &bookCount);
```

设计模式MVC

设计模式MVC		
model(模型层)—用来装载文件数据	Book.h	FileOpt.h
view(视图/界面层)—显示	BookView.h	
controller(控制层)	BookBiz.h	

Book.h——图书结构定义（所有其他头文件都含有）
FileOpt.h——文件操作(包含对文件的一些增删改查)
BookBiz.h——业务功能
BookView.h——图书的打印、菜单

- 设计步骤：模型层—界面层—控制层

目录：

一、操作时间

二、文件操作——FileOpt.h（含Book.h）

三、视图层——BookView.h（含Book.h）

四、业务层——BookBiz.h（含Book.h，FileOpt.h，BookView.h）

一、操作时间

步骤：

- 定义time_t类型的时间变量
- 3种时间操作方式
- 打印时间

（1）**time_t类型**

time_t实质上为long型，只是为long型定义了一个别名叫time_t。

time_t类型作为操作时间函数的固定类型（参数类型、返回类型）

```
time_t timet;    //定义time_t类型
```

（2）操作时间的相关函数

- **time()函数**

作用：计算1970-1-1到当前为止的秒数

调用方式(2种)：引用传参数，改变参数值。

调用空函数，并返回时间值。

```
time(&timet);        //方式一：引用传参数，改变参数值
timet=time(NULL);    //方式二：调用空函数，并返回时间值
```

- **localtime()函数**

调用方式：传入time_t指针，返回tm结构体指针

特点：针对“struct tm”结构体

年份从1900开始；月份从0开始

```
struct tm * pTime=localtime(&timet);
```

```c
time_t timet;

//操作时间有3种方式：
//1-time引用传参数，改变参数值
time(&timet);       //对timet进行时间赋值
//2-time调用空函数，并返回时间值
//timet=time(NULL);

printf("%d\n",timet);
printf("year:%d\n",timet/60/60/24/365 + 1970);  //当前大致年份

//3-针对“struct tm”结构体，调用localtime函数
struct tm * pTime=localtime(&timet);
printf("%d-%d-%d\n",pTime->tm_year + 1900,pTime->tm_mon + 1,pTime->tm_mday);
```

二、文件操作——FileOpt.h（含Book.h）

```
#define BookInfoPath "BookInfo.dat"  //定义全局常量
```

意义：在当前的项目目录下，建立一个"BookInfo.dat"文件，以后所有对文件的操作都针对该文件进行，所有图书信息也将保存到该文件中。

思考需求步骤：

增加图书：添加图书到文件（图书信息写入文件）——SaveBook

读取图书：将文件中的图书以数组的形式读出（从文件中取出图书信息）——GetAllBooks

构建数组需要：构建图书数组前，需要先知道图书总数（得到文件中图书总数）——GetBookCount

打印测试：打印文件中所有图书信息——printAllBooks

查找图书——GetBookById

删除图书——DeleteBookById

更新图书——UpdateBook

额外需求：获得文件中最后一本图书id+1——GetNewBookId

（1）void SaveBook(Book * book);

- 功能：将传入的单本图书追加保存到文件中（增加）
- 特点：用指针传参，可以方便在以后的操作中对该图书直接进行修改
- 函数设计步骤：
  - 判断传入图书指针是否为空
  - 将该图书内容追加保存到文件中（文件打开+文件读入）
  - 判断是否添加成功

（2）Book * GetAllBooks(Book * book, int * bookCount);

- 功能：从文件中获取所有的图书信息，返回到图书数组book中
- 特点：设计原理类似于操作时间的time()函数

- 参数1(Book * book)设计时为空，最后结果会返回该处；
- 参数2（int * bookCount）事先不用计算，只需定义名称，函数内部会计算并返回值。——通过指针的形式相当于多增加了一个返回值到该处

● 函数调用方式(2种)：

方式一：引用传参数，改变参数值。

方式二：调用空函数，并返回数组指针

```
int bookCount = -1;//图书总数
Book * bookArray = NULL;

//方式二：
bookArray = GetAllBooks(NULL, &bookCount);
//方式一：
//GetAllBooks(bookArray, &bookCount);
```

注意：当将函数修改为，只有传入参数1为null，才构建图书数组时，不能采用如下的调用形式：

Book * booArray = GetAllBooks(booArray, &bookCount);

● 函数设计步骤：

- 获得图书总数（调用GetBookCount函数）
- 判断传入图书指针是否为空，构建图书数组
- 将文件中的图书信息读入到图书数组中（文件打开+文件读出）
- 判断是否读取成功

（3）int GetBookCount();

● 功能：获得文件中包含的图书信息总数

● 函数设计步骤：

- 判断文件是否为空，若为空就返回0（文件打开）
- 计算总图书数量，当前一共的字节数/每个图书结构的字节数（文件指针移动）

（4）void printAllBooks();

● 功能：打印文件中的所有图书信息

● 特点：图书的借出时间打印

● 函数设计步骤：

- 将文件中的图书信息读入到图书数组中（调用GetAllBooks函数）

- 判断图书数组中是否为空
- 按Book结构对数组进行打印

（5）Book * GetBookById(int id);

- 功能：(查询) 根据图书的id，在文件中查找，并返回对应的图书信息。

  如果查到，返回该图书指针；如果没有查到，返回NULL

- 函数设计步骤：

- 获得图书总数（调用GetBookCount函数）
- 判断传入图书总数是否为空
- 用for循环：将文件中的每本图书信息读入到临时图书中，并与指定id进行比较判断

（6）int DeleteBookById(int id);

- 功能：(删除) 根据图书的id，删除文件中对应的图书信息

  如果删除成功，返回1；如果失败，返回0

- 特点：返回数字，为了对删除成功与否进行判断（也可以打印提示语句）
- 函数设计步骤：

- 将文件中的图书信息读入到图书数组中（调用GetAllBooks函数）
- 在图书数组中进行删除操作**（关键）**

① 要删除的是第一个（第一个为数组名，若改变，则不再是原来的数组），构建临时数组(长度-1)存放除第一个的其他

② 要删除的是最后一个(将长度-1，即可)

③ 要删除的是中间位置(先从当前位置，将后一个与前一个进行交换，再将长度-1)

④ 将删除后的结果都复制到临时数组数组中

- 将临时数组中的所有图书以覆盖的方式写回到文件中

（7）int UpdateBook(Book * book);

- 功能：(更新) 根据传入图书对应的id，更新文件中的某一本图书信息

  如果更新成功，返回1；如果更新失败，返回0

- 特点：

- 进行更新时，其他信息都能改变，只有id是确定不变的。
- 将文件指针置于当前书籍的开始处，以便覆盖：fseek(file, i*sizeof(Book), SEEK_SET);

- 函数设计步骤：

- 判断：如果没有传入需要更新的书数组指针NULL，或者当前文件中的图书数量为0（获得图书总数-调用GetBookCount函数），则更新失败
- for循环：按id进行查找要更新的这本书。
- 一旦找到，以覆盖的方式替代原来的书，写入该本书

（8）int GetNewBookId();

- 功能：获得文件中最后一本书的id+1
- 特点：移动文件指针,到最后一本书的开头，方便直接读最后一本书

  fseek(file, (bookCount-1) * sizeof(Book), SEEK_SET);

- 函数设计步骤：

- 获得图书总数（调用GetBookCount函数）
- 判断传入图书总数是否为0
- 移动文件指针,到最后一本书的开头（方便直接读最后一本书）
- 返回最后一本书的id+1，即可

三、视图层——BookView.h（含Book.h）

包含了图书的打印、菜单

登录菜单——主菜单——指定图书数组的图书信息

特殊需求：根据借阅状态，打印数组中相应的图书数组

（1）int ShowLoginMenu();

- 功能：打印登录菜单，并返回用户的选择

```
欢迎使用Extreme Acadamy图书管理系统
1、登录系统    2、退出
请选择:
```

- 函数调用：

```
int choice = ShowLoginMenu();
if (choice != 1) {
    exit(EXIT_SUCCESS); //不进行退出，就是成功登录了
}
```

（2）int ShowMainMenu();

- 功能：打印主菜单，并返回用户的选择
- 函数调用：choice = ShowMainMenu();

```
1、增加
2、删除
3、查找
4、借出
5、归还
欢迎进入图书管理系统，请选择您想要进行的操作:
```

（3）void PrintBookArray(Book * bookArray, int bookCount);

- 功能：打印传入的图书数组(bookArray)的前面(bookCount)个图书信息
- 函数调用：

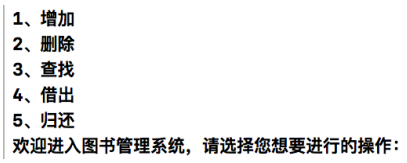
```
//打印全部图书信息
int bookCount = -1;
Book * book = NULL;
book=GetAllBooks(NULL, &bookCount);
PrintBookArray(book,bookCount);

//打印指定一本图书信息
Book * book1 = GetBookById(2);
PrintBookArray(book1, 1);
```

打印全部图书信息也可以直接调用函数：printAllBooks();

四、业务层——BookBiz.h（含Book.h，FileOpt.h，BookView.h）

核心应用操作：（整合-文件的交互、界面的显示）

增加-删除-查找-借出-归还



（1）void AddBook();

- 功能：由用户录入新图书信息，增加到文件中（调用SaveBook函数）
- 特点：id为自动生成(调用GetNewBookId函数)，时间自动生成当前时间

（2）void DeleteBook();

- 功能：由用户录入需要删除图书的id，从文件中删除（调用DeleteBookById函数）

（3）void SearchBook();

- 功能：由用户输入需要查找图书的id，从文件中查找出（调用GetBookById函数），并将相应信息打印出来（调用PrintBookArray函数）
- 特点：id为自动生成(调用GetNewBookId函数)，时间自动生成当前时间

（4）void BorrowBook();　　/**借出*/

- 功能：由用户输入需要借阅图书的id，从文件中查找出（调用GetBookById函数），修改文件的借阅状态（调用UpdateBook函数）
- 特点：可以显示出当前可以操作的图书列表，给用户一个参考
- 函数设计步骤：

- 打印可借图书列表
- 请输入要借出的图书编号，并进行查找
- 判断该图书的状态是否可借
- 若可借，将修改图书借出时间信息，覆盖到文件中

（4）void BorrowBook();　　/**归还*/

- 功能：由用户输入需要借阅图书的id，从文件中查找出（调用GetBookById函数），修改文件的借阅状态（调用UpdateBook函数）
- 特点：与借出BorrowBook()函数差不多
- 函数设计步骤：

- 打印可以归还图书列表
- 请输入要归还的图书编号，并进行查找
- 判断该图书的状态是否可借
- 若不可借（需要归还），将修改图书借出时间信息，覆盖到文件中

main.c中的主要调用程序如下：

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
#include "Book.h"
#include "FileOpt.h"
#include "BookBiz.h"
#include "BookView.h"

int main(int argc, const char * argv[]) {

    int choice = ShowLoginMenu();
    if (choice != 1) {
        exit(EXIT_SUCCESS); //不进行退出，就是成功登录了
    }

    while (1) {
        choice = ShowMainMenu();
        switch (choice) {
            case 1:
                //增加
                AddBook();
                break;
            case 2:
                //删除
                DeleteBook();
                break;
            case 3:
                //查找
                SearchBook();
                break;
            case 4:
                //借出
                BorrowBook();
                break;
            case 5:
                //归还
                RevertBook();
                break;
            case 6:
                //查看图书系统的所有图书
                printAllBooks();
                break;
            default:
                exit(EXIT_SUCCESS);
        }
    }

    return 0;
}
```