

**NANCHANG UNIVERSITY**

**C语言工程实训报告**



**题 目：** 图书馆管理系统

**学 院：** 软件学院

**专业班级：** 软件工程152

**姓 名：** 周寒聿西

**学 号：** 8000115057

**起讫日期：** 2016. 2. 29~2016. 3. 20

**指导老师：** 胡军

**摘要：**图书馆管理系统实现对图书馆日常各方面的管理自动化，减少图书馆工作管理人员的工作量，也方便读者使用。系统分为五个模块，实现对图书流通借阅情况的管理，对图书的管理，对图书分类的管理，对读者账户的管理，对管理员账户的管理。

**关键词：**C语言；图书馆管理系统

**引言**

为了实现对图书馆的管理自动化，减少图书馆工作人员的工作，这次工程实训实现了一个图书馆管理系统。这个系统分为读者信息管理模块，分类目录管理模块，图书编目管理模块，图书流通管理模块，管理员账户管理模块五个模块，具体功能包括完成与图书借阅相关的各种操作，查询图书信息和读者个人信息及借阅信息，实现工作人员更新图书信息、增加删除图书及分类、对读者账户信息进行操作、对管理员账户进行操作的功能。作者能力有限，有何错误疏漏之处，还望指正。

目录

[一、问题描述 4](#_Toc446262992)

[二、分析与设计 4](#_Toc446262993)

[1.1功能描述 4](#_Toc446262994)

[1.2程序流程图设计 5](#_Toc446262995)

[1.3数据结构设计 12](#_Toc446262996)

[1.4各模块实现方法 14](#_Toc446262997)

[1.5测试数据及结果 15](#_Toc446262998)

[三、源代码选摘 21](#_Toc446262999)

[四、总结 28](#_Toc446263000)

[4.1 可取之处 28](#_Toc446263001)

[4.2不足之处 29](#_Toc446263002)

# 一、问题描述

该项目是一个图书馆管理系统，需实现对图书馆日常各方面的管理自动化，减少图书馆工作管理人员的工作量，也方便读者使用。该项目要求能实现对图书流通借阅情况的管理，对图书的管理，对图书分类的管理，对读者账户的管理，对管理员账户的管理等等。

具体功能包括完成与图书借阅相关的各种操作，查询图书信息和读者个人信息及借阅信息，实现工作人员更新图书信息、增加删除图书及分类、对读者账户信息进行操作、对管理员账户进行操作的功能。

该项目要求能依据实际情况，对处理对象的各种信息进行相应的增删查改，并将最终确定的信息存入文件当中保存，也能将已保存在文件中的数据读出进行操作。

# 二、分析与设计

## 1.1功能描述

图书馆管理系统共分为五个模块:读者信息管理模块，分类目录管理模块，图书编目管理模块，图书流通管理模块，管理员账户管理模块。各个模块中都将有用的信息保存到相应文件中，也从相应文件中读取信息用于处理。

读者信息管理模块：该模块对读者的证件信息进行管理，完成新读者的注册增加，读者信息的修改、查询、浏览及读者证件挂失等功能。

分类目录管理模块：该模块对图书的分类进行管理，完成增加目录、删除目录、修改目录、以分类目录树的形式浏览目录等功能。

图书编目管理模块：该模块对图书的基本信息进行管理，完成新书信息的录入，被磨损书籍的淘汰，图书信息的修改、查询等功能。

图书流通管理模块：该模块对图书借阅的有关情况进行管理，完成图书的借阅、归还、续借、超期未还时罚款，图书的分类浏览，图书的检索，借阅记录的查询等功能。

管理员账户管理模块：该模块对系统管理员的账户进行管理，完成管理员账户的注册及删除，账户信息的查询、修改，管理员账户密码的修改等功能。

## 1.2程序流程图设计

系统整体流程图

显示主菜单

选择

流通管理

退出系统

图书编目

分类管理

读者管理

显示子菜单

选择

操作

读者管理模块

增加读者

打开文件

将读者信息导入链表

否

是否成功

寻找链表尾节点

是

新建节点录入信息并保存

否

录入是否结束

是

查询读者

将读者信息导入链表

输入读者证件号

从链表取一条记录

是

是否匹配

输出结果

否

否

是否已到链表尾

是

输出“未找到记录”

修改读者

将读者信息导入链表

输入证件号

查找链表

否

是否找到

是

修改记录

保存记录

分类目录管理模块

修改分类 输出分类目录树

打开文件

输入分类编号

建立双亲储存结构

打开文件

输出根节点

fread读取一条记录

检索孩子节点

比较关键字

否

是否有孩子

否

是

匹配

否

是第一个？

是

fseek定位指针

是

level++

fwrite修改 记录

图书编目管理模块

新书录入 删除图书

打开文件

输入ＩＳＢＮ

建立链表

是

是“#”？

输入ＩＳＢＮ

否

输入其他字段

查找记录

建立链表节点，

插入链表

否

找到？

是

保存链表

删除节点

保存链表

图书流通管理模块

　　　　借书　　　　　　　　　　　　　　　还书

生成还书时间

获取系统时间

输入证件号

计算应还时间

输入证件号

输入ＩＳＢＮ

是

查找借阅记录

否

挂失？

修改借阅记录

是

借满１５本？

否

保存借阅记录

输入ＩＳＢＮ

继续还书？

保存借阅记录

是

否

否

是

继续借阅

　　　　　　　　 续借

输入证件号

输入ISBN

查找借阅记录

获取系统时间

修改借阅记录

保存借阅记录

是

继续续借？

否

账户管理模块

注册账户

打开文件

否

输入注册信息

是

用户名唯一？

保存账户信息

关闭文件

修改密码

打开文件

读取记录

输入用户名

是

查询记录

继续？

否

找到？

是

否

输入新密码1次

输入新密码2次

否

相同？

是

保存账户信息

关闭文件

## 1.3数据结构设计

（１）读者信息

读者的个人信息包括读者证证件号，读者姓名、性别、工作单位和证件状态，建立一个结构体类型来储存单个读者的这一系列信息，以单链表的形式储存在内存中。

结构体的具体内容如下：

typedef struct ReaderLink

{

char code[20];

char name[30];

char sex[5];

char dept[50];

short tag;

struct ReaderLink \*next;

}ReaderLink;

（２）分类目录

分类目录的信息包括该分类的编号、名称及其父类编号，用一个结构体类型保存这些信息。因为图书的分类是分层的，所以采用树形结构储存分类目录信息。

结构体的具体内容如下：

typedef struct Catalog

{

char key[20];

char caption[80];

char parent[20];

}Catalog;

树形结构的结构体具体内容如下：

typedef struct CatalogTree

{

Catalog Node[MAX\_TREE\_SIZE];

int root;

int num;

}CatalogTree;

（３）图书信息

图书信息包括书名、ISBN编号、作者、出版社、出版日期、标价和分类编号，也用一个结构体类型保存，以顺序表的形式储存在内存中。

结构体具体内容如下：

typedef struct Book

{

char ISBN[50];

char title[200];

char author[200];

char publisher[100];

char date[20];

float price;

char CatalogNum[200];

}Book;

顺序表结构体如下：

typedef struct BookList

{

Book \*Node;

int Num;

}BookList;

（４）借阅信息

借阅信息包括被借书籍的ISBN编号、借阅者证件号、借书日期、应还日期、归还日期、借阅状态和罚款，仍然用一个结构体储存这些信息，以单链表形式储存在内存中。

结构体具体内容如下：

typedef struct LendInfo

{

char code[20];

char ISBN[50];

char LendDate[20];

char LastDate[20];

char ReturnDate[20];

float fine;

int tag;

}LendInfo;

单链表结构体如下：

typedef struct LendInfoLink

{

char code[20];

char ISBN[50];

char LendDate[20];

char LastDate[20];

char ReturnDate[20];

float fine;

int tag;

struct LendInfoLink \*next;

}LendInfoLink;

（５）管理员账户信息

管理员账户信息包括用户名，姓名，密码，工号，用结构体储存数据，以单链表为储存结构。结构体具体内容如下：

Typedef struct AdminLink

{

char Username[20];

char Name[20];

int JobNO;

char Password[20];

struct AdminLink \*next;

}AdminLink;

## 1.4各模块实现方法

1. 读者管理模块

应用顺序文件操作及对链表的相关操作。多次用顺序文件操作从文件中逐条读入信息到链表，对链表或其节点输出、修改数据，将数据存入新增节点，删除节点，链表中信息有改动的操作后，用顺序文件操作将链表中的数据写入文件中保存。注意遍历链表时循环控制条件的把握。

1. 分类目录管理模块

运用随机文件操作及对树的相关操作。多次用随机文件操作对分类信息进行读写，在树形结构中对结点进行增删查改，甚至对树形结构进行改变，运用递归和回溯法将分类目录树输出到屏幕。

1. 图书编目管理模块

运用顺序文件操作将图书信息读入到链表中，用字符串的相关操作对图书信息进行修改，对链表增加、删除节点以完成新书录入和图书删除。

1. 图书流通管理模块

运用顺序文件操作将图书信息读入到顺序表中，运用排序方法对图书进行排序，之后用字符串的相关操作及折半查找的方法依照相应要求查找图书，输出图书信息，完成图书浏览；将借阅记录读入到顺序表中，运用查找方法找到借阅信息并输出；运用时间处理函数生成借书、应还日期及时间差，并以做标记的方式完成借阅、归还、续借操作，对归还日期和应还日期的时间差进行计算，得出罚款金额。

1. 账户管理模块

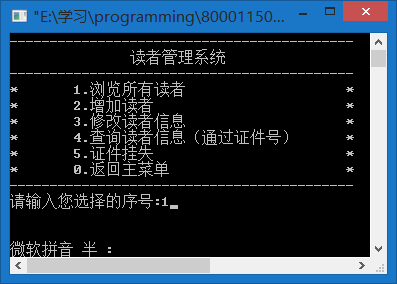
运用随机文件操作及对结构体的相关操作，完成新账户的建立，账户信息及密码的修改和账户的删除。此模块中一次只对一个账户进行操作，且账户之间相关性不强，所以不用单链表或顺序表等存储结构也能完成要求功能。

## 1.5测试数据及结果

**主菜单：**



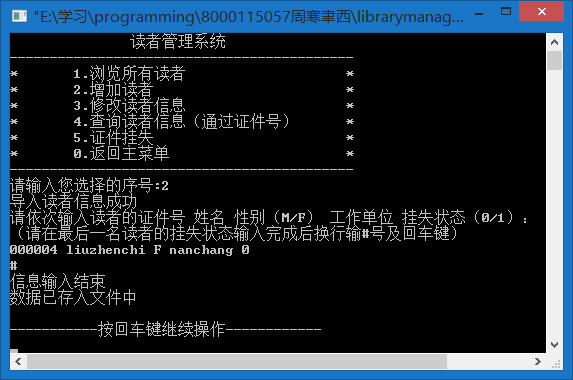
**读者管理系统：**



浏览所有读者：



增加读者：

****

修改读者信息：

****

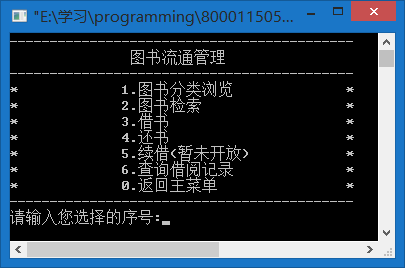
查询读者信息（通过证件号）：

****

证件挂失：

****

**图书流通管理系统：**

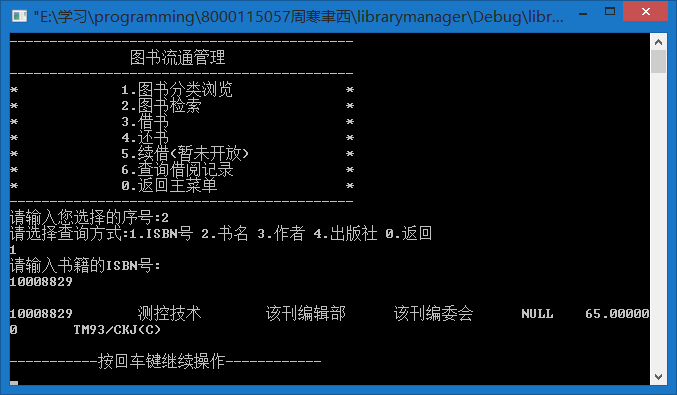
****

图书分类浏览：

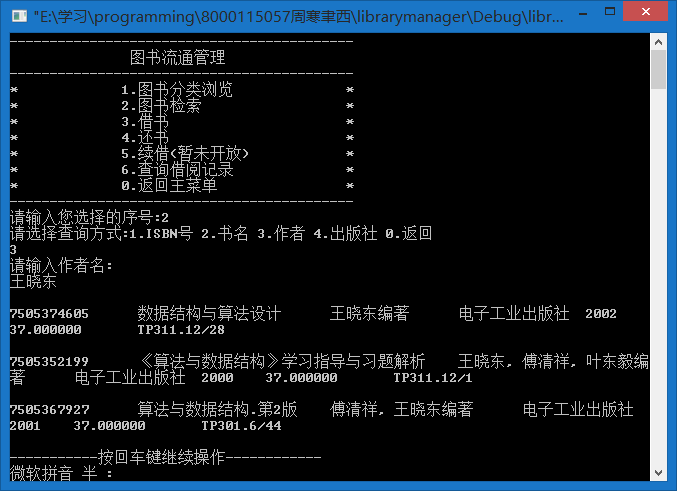
****

图书检索（只选取部分测试结果）：

按ISBN号：

****

按作者：

****

借书：

****

还书：

****

查询借阅记录：

****

# 三、源代码选摘

**增加读者**

void InputReader()

{

int flag=1;

char code[20];

ReaderLink \*head,\*p,\*s;

FILE \*fp = NULL;

head = CreatReaderLink();//将读者信息导入链表中

fp = fopen("reader.txt","a");

if(fp == NULL)

{

printf("文件打开失败\n");

ReaderMenu();

}else{

printf("请依次输入读者的证件号 姓名 性别（M/F） 工作单位 挂失状态（0/1）：\n");

printf("（请在最后一名读者的挂失状态输入完成后换行输#号及回车键）\n");

//寻找尾结点

p = head->next;

while(p->next != NULL)

{

s = p;

p = s->next;

}

//录入信息

while(flag)

{

scanf("%s",code);

if(strcmp(code,"#") != 0)

{

s = (ReaderLink \*)malloc(sizeof(ReaderLink));

p->next = s;

p = s;

p->next = NULL;

strcpy(p->code,code);

scanf("%s %s %s %1d",p->name,p->sex,p->dept,&p->tag);

fprintf(fp,"%s\t%s\t%s\t%s\t%1d\n",p->code,p->name,p->sex,p->dept,p->tag);

}else{

flag = 0;

printf("信息输入结束\n");

}

}

printf("数据已存入文件中\n");

fclose(fp);

}

}

**证件挂失**

void LostReader()

{

char Code[20];

int flag = 1;

ReaderLink \*head,\*p,\*s;

head = CreatReaderLink();

printf("请输入要进行挂失的证件号:");

scanf("%s",Code);

p = head->next;

while( p != NULL)//遍历链表

{

if(strcmp(Code,p->code) == 0)//找到需挂失的证件号

{

p->tag = 1;//挂失操作

printf("挂失成功！");

flag = 0;

break;

}else{

s = p->next;

p = s;

}

}

if(flag)

{

printf("未找到输入的证件号\n");

}else{

SaveReader(head);

}

}

**保存读者信息**

void SaveReader(ReaderLink \*head)

{

ReaderLink \*p,\*s;

FILE \*fp = NULL;

//打开文件

fp = fopen("reader.txt","w");

if(fp == NULL)

{

printf("文件打开失败，信息未保存\n");

}else{

//输出信息到文件

p = head->next;

while(p != NULL)

{

fprintf(fp,"%s\t%s\t%s\t%s\t%1d\n",p->code,p->name,p->sex,p->dept,p->tag);

s = p->next;

p = s;

}

printf("读者信息已保存到文件中\n");

//关闭文件

fclose(fp);

}

}

**输出分类目录树**

CatalogTree CreatCatalogTree()

{

int i = 0;

FILE \*fp;

CatalogTree C;

if((fp = fopen("catalog.dat","rb")) == NULL)

{

printf("文件打开失败，无法导入分类信息\n");

}else{

fread(&C.Node[i],sizeof(Catalog),1,fp);//读取分类目录信息

while(!feof(fp))

{

i++;

fread(&C.Node[i],sizeof(Catalog),1,fp);

}

C.num = i;

C.root = 0;

printf("分类信息导入成功\n");

fclose(fp);

PrintCatalogTree(C);//输出分类目录树

return C;

}

}

void PrintCatalogTree(CatalogTree tree)

{

int level = 1;

PreOrder(tree,0,level);

}

//先根遍历

void PreOrder(CatalogTree tree,int k,int level)

{

int i,j,m = 0;

for(i=1; i<level; i++)

{

printf(" ");

}

printf("|---%s %s\n",tree.Node[k].caption,tree.Node[k].key);//输出父节点

for(j=k+1; j<tree.num; j++)

{

if((strcmp(tree.Node[j].parent,tree.Node[k].key)) == 0)

{

if(m == 0)

{

PreOrder(tree,j,++level);//输出第一个孩子节点

m ++;

}else{

PreOrder(tree,j,level);//输出其他孩子节点

}

}

}

}

**借书**

void LendBook()

{

int flag=0;

int booknum;

char ISBN[50];

ReaderLink \*p;

BookList B;

Book book;

LendInfo L;

char LendDate[20];char LastDate[20];

time\_t t;

t = time(NULL);

strftime(LendDate,sizeof(LendDate),"%Y-%m-%d",localtime(&t));//获取借阅时间

t = t + MAX\_DAY\_LEND\*24\*60\*60;

strftime(LastDate,sizeof(LastDate),"%Y-%m-%d",localtime(&t));//获取应还时间

p = FindReader\_code();//找到该读者

if(p == NULL)

{

flag = 1;

}else{

booknum = LendBookNum(p->code);

if(p->tag == 1)

{

printf("该证件已挂失，无法借阅\n");

flag = 1;

}else if(booknum > MAX\_BOOK\_LEND)

{

printf("已经借满%d本，无法再借阅\n",MAX\_BOOK\_LEND);

flag = 1;

}else{

printf("-----------图书借阅，输入#号及回车结束----------\n");

printf("请输入书籍的ISBN:\n");

scanf("%s",ISBN);

while((strcmp(ISBN,"#") != 0) && (booknum < MAX\_BOOK\_LEND))

{

B = CreatBookList();

B = SelectSort(B);

book = SearchISBN(B,ISBN);

if(strcmp(book.ISBN,ISBN) == 0)//找到该书

{

strcpy(L.code,p->code);

strcpy(L.ISBN,ISBN);

strcpy(L.LendDate,LendDate);

strcpy(L.LastDate,LastDate);

strcpy(L.ReturnDate,"NULL");

L.tag = 0;

printf("%s\t%s\t%s\t%s\t%1d\n",L.code,L.ISBN,L.LendDate,L.LastDate,L.tag);

SaveLendInfo(L);//保存借阅信息

printf("-----------图书借阅，输入#号及回车结束----------\n");

printf("请输入书籍的ISBN:\n");

scanf("%s",ISBN);

}

}

flag = 0;

}

}

if(flag)

{

CirculationMenu();

}else{

printf("借阅完毕\n");

}

}

**还书**

void ReturnBook()

{

char ISBN[50];

char Code[20];

char ReturnDate[20];

int flag = 0;

time\_t t;

LendInfoLink \*head,\*p,\*s;

FILE \*fp = NULL;

head = CreatLendInfoLink();//将借阅信息导入到链表

t = time(NULL);

strftime(ReturnDate,sizeof(ReturnDate),"%Y-%m-%d",localtime(&t));//生成还书时间

printf("请输入证件号:\n");

scanf("%s",Code);

printf("-----------图书归还，输入#号及回车结束----------\n");

printf("请输入书籍的ISBN:\n");

scanf("%s",ISBN);

if(strcmp(ISBN,"#") != 0)

{

while(strcmp(ISBN,"#") != 0)

{

p = head->next;

while(p != NULL)

{

if(strcmp(p->code,Code) == 0)//找到该读者

{

if(strcmp(p->ISBN,ISBN) == 0)//找到该书

{

strcpy(p->ReturnDate,ReturnDate);

p->tag = 1;

printf("证件号\tISBN\t借书日期\t应还日期\t归还日期\t借阅状态\n");

printf("%s\t%s\t%s\t%s\t%1d\n",p->code,p->ISBN,p->LendDate,p->LastDate,p->tag);

flag = 1;

break;

}

}

s = p;

p = s->next;

}

if(flag == 0)

{

printf("证件号与书籍不匹配!\n");

}

flag = 0;

printf("-----------图书归还，输入#号及回车结束----------\n");

printf("请输入书籍的ISBN:\n");

scanf("%s",ISBN);

}

fp = fopen("lends.txt","w");

if(fp == NULL)

{

printf("文件打开失败,无法保存归还记录\n");

fclose(fp);

CirculationMenu();

}else{

//保存还书记录

for( p=head->next; p!=NULL; p=p->next )

fprintf(fp,"%s\t%s\t%s\t%s\t%d\n",p->code,p->ISBN,p->LendDate,p->LastDate,p->tag);

fclose(fp);

printf("借阅记录已成功保存到文件\n");

}

}

printf("还书结束\n");

}

# 四、总结

在这次实训中，未能完成整个图书馆管理系统，只完成了读者管理系统和图书流通管理系统的大部分功能，其他部分只了解大概的实现方法，没有进行实际代码实现。这次实训是第一次要规划一个复杂系统的结构，一开始深感自己的全局观不够强，考虑做不到全面，在实训中，这种能力得到了提高，算是体会了一次自顶向下对一个系统的规划，函数也能尽量写得独立性、可移植性强一些，方便在不同情况下多次调用。

## 4.1 可取之处

读者管理系统中，主要运用顺序文件操作及链表有关操作。顺序文件操作避免最后一条数据的重复读取，在循环前先读取一条数据。链表的遍历过程中节点的移动及尾节点的判定也很顺畅。

图书流通管理系统中，主要运用顺序文件操作、时间处理函数及排序、查找、递归回溯等算法完成各项功能。其中图书分类浏览功能中运用递归和回溯法将分类目录树输出，递归的运用难度比较大，完成得较好。借书还书功能中，时间处理函数之前没有运用过，其中时间的各种格式转换比较复杂，完成得较好。

每次文件读写时，若打开文件失败，《C语言工程实训教程》中的代码都是让程序立即终止，在作业中，我将其改为回到最近一级菜单，让用户有更好的使用体验。

## 4.2不足之处

（1）在应该完成的两个模块中，有一些完成得不好之处。

图书流通管理系统的续借和罚款功能没能来得及完成，可在借阅记录的结构体中加上借阅、归还时的机器日历时间、续借次数及罚款金额等成员，成员信息保存到借阅记录的文件中，办理还书或续借手续时，计算借还时间差，差值在规定范围内才允许归还，续借则还要求续借次数未超过上限，否则依据计算出的时间差给出相应罚款数，缴纳罚款后才可继续还书或进行下一次借书。

（2）在各个模块中，还有一些教程中也不完美的地方。

读者信息中可加上读者的借阅情况，包括图书到期未还次数、损坏图书的记录、罚款次数及罚款缴纳情况，这些记录由系统根据读者的书籍借还及罚款情况自动生成并保存在读者信息文件中。在读者管理模块或图书流通管理模块中可加入对读者借阅情况的查询，可方便读者和管理员对借阅情况的了解。并且罚款缴纳情况为未缴纳的读者无法再进行借书。

图书信息中可加上该图书的借阅次数、借阅状态和对该图书的简介，借阅次数由系统依据读者对该书的借阅情况自动生成，图书简介可由工作人员在新书录入时一并录入。在图书流通管理系统的图书检索功能中，可加上对图书简介的输出，使读者在检索图书时对该书的内容有所了解；在借书过程中，借阅状态为已借出的图书不能再被借阅；在图书编目管理系统的查询图书功能中，可加上图书借阅次数的输出，使管理员对该书的借阅情况有所了解，可以根据图书受欢迎的程度决定是否增加该书的馆藏数量。

除图书流通模块的其他模块均为工作管理人员所用，含有较为重要的信息，不能随意被人改动，可以在进入这些模块时要求登陆，输入密码，使有权限者才能进入其中对数据进行操作。

分类目录管理模块中增加分类时，应根据新分类的编号确定该分类信息在文件中的存储位置，因为新分类可能不属于文件末的父类，所以要找到新分类所属的新的父类，再将信息写入该处；修改分类时也一样要找到该分类所属的新的父类，再将新信息写入该处。

**参考文献：**

[1] 《C语言工程实训教程》 王卓，万立中，刘伯成 人民邮电出版社 2013年12月