**图书管理系统程序说明文档**

学院：计算机学院

专业：计算机科学与技术

组别：嵌入式组

姓名：钟惠明

目录

[项目简介 1](#_Toc513029341)

[总体设计 1](#_Toc513029342)

[详细设计 3](#_Toc513029343)

[一、 入库 3](#_Toc513029344)

[二、 清除 6](#_Toc513029345)

[三、借阅 7](#_Toc513029346)

[四，归还 10](#_Toc513029347)

[五、查看 13](#_Toc513029348)

[六、菜单 16](#_Toc513029349)

[数据结构 18](#_Toc513029350)

[程序亮点 21](#_Toc513029351)

[程序兼容性说明 21](#_Toc513029352)

[总结与体会 21](#_Toc513029353)

# 项目简介

本项目为图书管理系统，主要用于存放图书信息和记录学生的借阅情况等。

主菜单会提供管理员菜单和用户菜单两个选项，管理员需要密码，学生则需要输入学号。其中管理员可以将图书入库，也可以清除库存、查看所有借书记录和黑名单。用户可以借还书籍和查看自己的借阅情况，同时管理员和用户均可查看库存的所有图书及其信息。

一个学生一次最多可借5本书，超过30天未归还视为逾期，逾期超过三次就会被拉入黑名单（但这里没有给予相应的处罚）。

# 总体设计

本项目使用了三条主链表，分别用来存放图书全部信息、图书借阅信息和黑名单信息。三条主链表的头指针均为全局变量。其中存放图书信息的主链表结点只存放图书类型和指向该类型书籍的次级指针，次级链表结点用于存放书籍信息，包括书名、编号、作者、类型、现存量和总存量等。

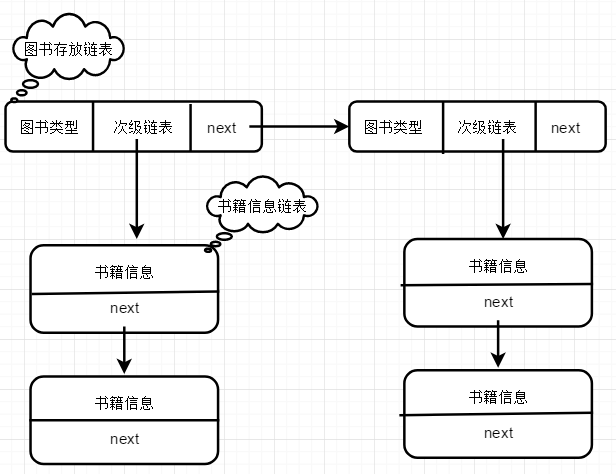
图书借阅链表也使用了两层链表，其中主链表记录结点存放的是学生的学号、姓名、逾期归还次数、当前借书总量和和次级链表头指针。次级链表书籍结点存放着书名、借阅日期、最迟归还日期和指向书信息的指针等。

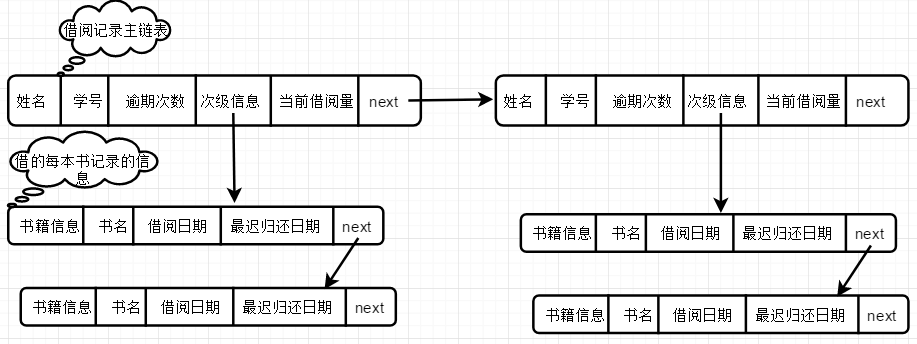
黑名单链表用于存放学生的学号、姓名和逾期次数。

一个学生一次最多可借5本书，当该学生的当前借书量超过5本时，系统会给出提示并且不借书给他。当书籍超过30天未归还视为逾期，逾期超过三次就会被拉入黑名单（但这里没有给予相应的处罚）。

主菜单会提供管理员菜单和用户菜单两项，其中管理员可以将图书入库，也可以清除库存、查看所有借书记录和黑名单。用户可以借还书籍和查看自己的借阅情况，同时管理员和用户均可查看库存的所有图书。所有操作均在输出窗口完成。根据管理员/用户的不同输入进行不同的操作，再将结果输出到窗口。

**整体结构示意图：**





# 详细设计

本项目总共分为入库、清除、借阅、归还、查看和菜单六个部分。

## 入库

入库部分有四个函数，分别为transform() 、searchKind() 、inLibrary()和input()，由input函数调用其他函数实现图书入库。这个部分用到两个全局变量M和N，用于记录入库的书的类型排位和同类型中的排位，通过这两个变量确定该书的编号。

函数名：void input( )

参数：无

功能：管理员菜单函数直接调用本函数，调用其他三个函数，实现图书入库。

流程图：



函数名：void inLibrary ( Book \*p，int n )

参数：记录书籍信息的结点p和书籍数量n

功能：通过调用serchKind找到该书的分类，再从该类型中查找，若之前已经有这书了，则直接增加数量，否则遍历到结尾添加该结点。

流程图：



函数名：Book \*searchKind(char kind ) ;

参数：书籍类型 kind ，返回值为指向该分类的第一本书的指针

功能：在图书主链表中查找该分类，若没有则在链表尾创建一个结点，返回指向该分类的第一本书的指针

流程图：



函数名：void transform( int m ,int n , char \*num )

参数 ：整数m 、n和字符指针num；

功能：将两个整数化为字符串放到指针num指向的数组中。

## 清除

整部分就一个函数，从图书系统中清除指定书名的书籍。

函数名：Status destroyBook( )

参数：无

功能：从系统中清除指定书名的书籍

流程图：



## 三、借阅

借阅部分也有四个函数，分别为borrow( ) 、addDate( ) 、searchBook ( )和searchStudent( ) 。其中borrow( )函数为总入口，由菜单函数直接调用，再由其调用另外三个函数，实现借阅功能。

函数名： BorrowRecord \*searchStudent( char \*studentNum)

参数：指向学生学号的指针，

功能：通过参数寻找对应的学生，若未找到则表示该学生第一次借书，则在链表尾新增结点指向该学生。返回指向该学生的记录指针。

流程图：



函数名：Book \*searchBook( char \*name)

参数：指向书名的指针name

功能：根据书名在图书存储链表找查找该书，返回指向该书的节点。

流程图：



函数名：void addDate (char \*date ，char \*returndate )

参数：一个指向借阅日期的指针和一个指向最迟归还日期的指针

功能：根据借阅日期推算最迟归还日期，将结果放到最迟归还日期指针指向的数组里

函数名：void borrow ()

功能：调用前面三个函数以实现借阅功能。

流程图：



## 四，归还

归还部分有三个函数，分别为searchRec ordStudent( ) 、addBlackList()和giveback( )函数，由giveback函数调用其他两个函数，完成整个归还部分，通过查看该学生是否逾期归还可将其拉入黑名单。

函数名： BorrowRecord \*searchRecordStudent ( char \* studentNum )

参数：指向学生学号的指针studetnNum

功能，在借阅记录中查找该学号的学生，返回指针他的指针，若未找到则返回空。

流程图：



函数名：void addBlacklist (BorrowRecord \*R )

参数：指向学生借阅记录的指针

功能：在黑名单记录链表中查找该学生，若找到则将他的逾期次数加1，若未找到则在链表尾新增节点，将其加入黑名单。

流程图：



函数名：void giveback()

功能：通过调用上面两个函数实现还书功能，同时判断是否逾期归还和逾期次数以决定是否将其加入黑名单

流程图：



## 五、查看

查看部分有四个独立的函数，分别为ViewRecord() 、ViewSelf( ) 、outputAllbook() 和LookBlackList()。可以提供管理员查看所有结束记录和黑名单，可以提供学生查看自己的借阅情况，管理员和学生均可以查看所有图书信息

函数名：void ViewRecord()

功能：提供管理员查看所有借阅记录

流程图：



函数名：void ViewSelf( )

功能：提供学生查看自己的借阅情况。

流程图：



函数名：void outputAllBook( )

功能：输出所有图书信息

流程图：



函数名：LookBlackList( )

功能：提供管理员查看黑名单

流程图：



## 六、菜单

这部分有四个函数，分别为输出主界面函数welcome( ) 、管理员菜单函数AdmanistrstorMenu( ) 、用户菜单函数StudentMenu( )和主菜单函数menu( )，通过三个菜单函数实现各个功能的选择与使用。

函数名：void welcome ()

功能：输出主界面窗口。

函数名：void AdministratorMenu( )

功能：调用各函数，提供管理员使用各项功能

流程图：



函数名：StudentMenu( )

功能：调用各函数以提供学生使用各项功能

流程图：



函数名：void menu( )

功能：开始界面，提供选择用户类型。用户类型为管理员和学生。

# 数据结构

本项目使用的数据结构主要为数组和链表。其中使用了三条链表，分别用于存放书籍、借阅记录和黑名单记录。每条链表的头节点都用全局变量固定。其中存放书籍和存放借阅记录都使用了两层链表。

**存放书籍信息的结构体：**

typedef struct book //每本书的结构体

{

char num[7]; //编号,每本书的编号前两位为类型排位，后四位为该书在该类型中的排位

char name[20]; //书名

char author[20]; //作者

char kind; //书的类型，这里定义书的类型为A,B,C....

int nowStock; //现存量

int sumStock; //总库存量

struct book \*next;

}Book;

//图书存放系统结构体，用于存放书籍的分类和指向他们的指针

typedef struct library

{

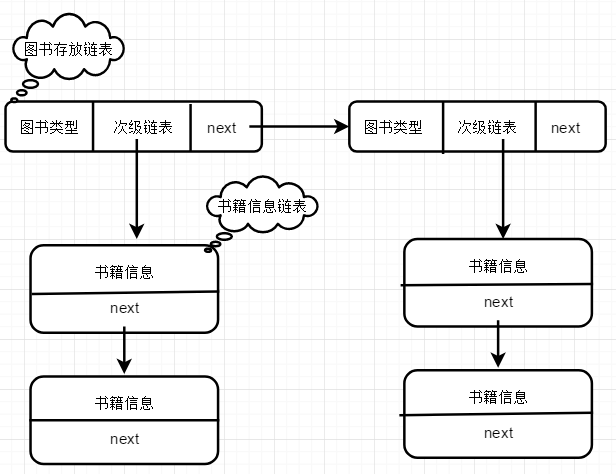
char book\_Kind;

Book \*bookHead;

struct library \*next;

}Library;

图书存放示意图：



**存放借阅记录的结构体：**

//借阅的书籍信息

typedef struct borrow

{

Book \*book; //指向书的信息

char bookName[20]; //书名

char date[10]; //借阅日期

char returndate[10]; //最迟还书日期

struct borrow \*next;

}Borrow;

//学生借书记录

typedef struct borrowRecord

{

char studentNumber[10];

char studentName[20];

int overdueTime; //逾期归还的次数，超过十次将拉入黑名单

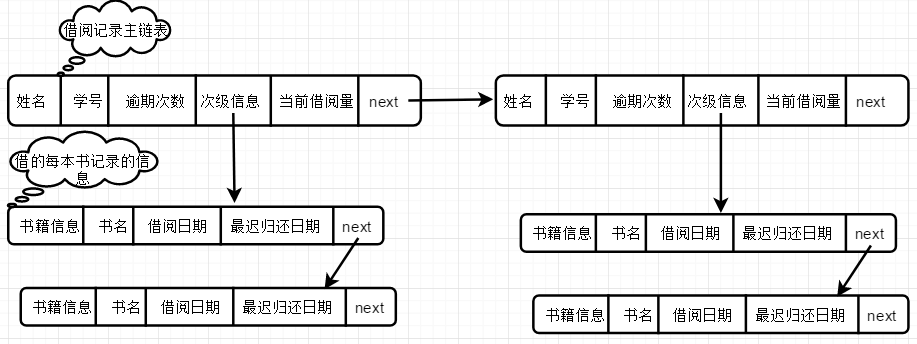
int bookNum; //当前借书总量，超过十本将不能借书

Borrow \*bookHead; //借的书

struct borrowRecord \*next;

}BorrowRecord;

借阅记录信息存放示意图：



**黑名单记录结构体：**

typedef struct blacklist //黑名单记录

{

char studentNumber[10]; //学号

char studentName[20]; //姓名

int overdueTime; //逾期次数

struct blacklist \*next;

}BlackList;

# 程序亮点

本程序除了完成了要求的内容以外，还增加了管理员查看黑名单和所有借阅记录以及学生查看本人的借阅情况等功能。

# 程序兼容性说明

本程序在使用过程中会限制输入的长度，以防止输入过长导致程序崩溃，当输入超过一定长度的时候，系统会输出相应的提示，并要求重新输入。在主窗口的旁边也有提示一些相关的信息。像输入的学号默认为8位，输入的日期也默认为8位，姓名、书名和作者等的默认长度均为20字符以内，当超过这个长度时就会提示出错。

图书类型也固定为26个大写英文字母，因为需要通过图书类型来确定书籍编号，但书籍类型是无序的，即26个字母随便先输入哪个，确定编号只是根据入库的先后顺序来的。一次入库数据数量限定在100本以内，若超过100本则需要二次输入。

本程序的缺陷部分为日期需要主动输入，无法自动获取当天的日期。

# 总结与体会

在这个图书管理系统项目中，我除了完成了基本的功能以外，还新增加管理员查看黑名单和所有借阅记录以及学生查看自己的借阅情况等三个功能。主要用到的数据结构为链表，全部功能实现都依赖于链表。所有的输入也有相应的限制和错误提示。

通过完成这次考核任务，我学会了去完成开发项目的一些必要步骤，包括说明文档、流程图和说明PPT，其中说明文档需要说清除什么内容等等，这些我之前并不太懂，只觉得代码可以让别人看懂就好了。现在通过完成这个图书管理系统，我认识到做一个项目其实并不是只敲代码就可以的，虽然代码是主要的，但还需要我们完成很多其他的东西。

在这次完成项目的过程中，我得到更多的是经验，写代码的经验、画流程图的经验、编辑文档的经验和制作PPT的经验。这几天我可以说是在高压状态下完成任务的，同时我也感觉到自己敲代码的能力比之前有了明显的进步，现在基本上不会出现什么语法错误了，像之前基本上都会出现一堆警告和错误提示的。同时我感觉到自己敲代码的效率也提高了不少，五天时间三天假，还有一大堆其他作业，以前我是根本不敢想象自己可以在这么短的时间内去完成一个项目的，但现在我竟然做到了，看到自己的进步心中还是有点小骄傲的。同时我也学会了去完成一份像样一点的说明文档，之前学习了编辑文档，现在正好可以练习使用，也收获了许多经验。