**《数据结构》课程设计**

**报告**

学 院: 专业班级:

学生姓名: 学 号: 2

设计地点（单位） 计算机自主学习中心

设计题目: 基于哈希表的图书馆管理系统

完成日期： 2022 年 6 月 24 日

指导教师评语: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

成绩（五级记分制）:\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

指导教师（签字）:\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

**重庆科技学院**

**课程设计任务书**

设计题目：基于哈希表的图书馆管理系统

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学生姓名 | |  | | |
| 课程名称 | | 数据结构课程设计 | 专业班级 |  |
| 地 点 | | 计算机自主学习中心 | 起止时间 | 2022.6.13-2022.6.24 |
| 设计内容及要求 | 某高校为更好的管理图书馆，现需设计一简易图书管理系统，实现新书录入，图书资料查询，显示总图书信息功能。要求开发的系统需要具备以下功能：  1.实现读者借阅图书；  2.实现读者归还图书；  3.实现对图书信息的综合管理：设置二级子目录实现信息的增，删，改，查等操作。  4.实现对读者信息的综合管理：设置二级子目录实现信息的增，删，改，查等操作。  5.显示用户信息：显示所有用户信息，含学号、姓名、借阅状况等信息；  6.返回主界面  进一步掌握利用C语言进行程序设计的能力；进一步理解和运用结构化程序设计的思想和方法；初步掌握开发一个小型实用系统的基本方法；学会调试一个较长程序的基本方法；学会利用简单的用例图表示需求分析；利用流程图表示算法。  根据题目要求，运用所学知识，完成设计和实现工作，并按照书写规范，撰写课程设计报告。 | | | |
| 设计参数 | 1. 系统至少有图书管理员、用户两种角色。 2. 系统提供控制台界面。 3. 支持从文本文件导入图书信息，导入用户信息。 4. 用二进制文件存储图书信息和用户信息。自定义文件格式。 5. 用文本文件保持日志文件。每天操作保持成一个日志文件。自定义日志文件格式。 6. 图书信息、用户信息从文件读入内存后，保存在哈希表中。 7. 可以增加删除图书，增加删除用户。 8. 可以按书名、作者查询图书；可以按学号、姓名查询用户。 9. 可以查询某一时间段内所有图书借阅情况，例如查询从2022年1月1日到2月13日之间所有被借阅的图书，并按借阅次数排序；例如查询从2022年5月9日到6月3日之间所有来借阅图书的同学，并按学号排序；或者按借阅次数排序。 10. 系统分析、设计正确 11. 界面简洁、美观。 12. 报告格式规范。 | | | |
| 进度  要求 | 6月13日 讲解课程设计任务  6月14日 阅读课程设计指导书  6月15日 实现栈与队列相关算法的代码  6月16日 实现二叉树相关算法  6月17日 理解二叉搜索树  6月20日 实现图的遍历  6月21日 实现最短路径算法  6月22日 实现排序问题的代码  6月23日 编写课程设计报告  6月24日 课程设计答辩 | | | |
| 参考资料 | 1．Mark Allen Weiss, 数据结构与算法分析：C语言描述，北京：机械工业出版社，2019.11  2．邓俊辉，数据结构(C++语言版)，北京：清华大学出版社，2013.9  3．K．N．King，C语言程序设计现代方法；北京：人民邮电出版社，2010.2 | | | |
| 其它 | 附加题：完成课程设计任务之后，可以学习CGI开发，用HTML页面作为系统界面，用C语言编写服务器脚本，实现一个有图形用户界面的图书管理系统。  学习资源1：<https://www.bilibili.com/video/BV1Q4411J7GN/>  学习资源2：<https://blog.csdn.net/gnefniu/article/details/42432657> | | | |
| 说明 | 1.本表应在每次实施前一周由负责教师填写二份，学院审批后交学院教务办备案，一份由负责教师留用。  2.若填写内容较多可另纸附后。3.一题多名学生共用的，在设计内容、参数、要求等方面应有所区别。 | | | |

系主任：葛继科 指导教师：伍建全

2022年6 月1日

# 摘要

编程课程是计算机类较为重要的专业课程，包括C语言，数据结构，算法设计和分析以及其他更重要的课程。目前，在这类课程的教学过程中，管理困难以及评价机制老化，因此，本系统计划利用本学期所新学的数据结构课程内容与个人经验记录，完成一个图书馆管理系统。

本文阐述了基于图书管理系统的设计与开发，本系统基于C语言实现，使用哈希表，链表等数据结构将用户、图书、借阅信息存储到该系统中，该系统还拥有导出图书用户和借阅记录等功能。系统还具有操作日志功能，对用户的增删查改以及图书的增删查改都有所记录。

图书馆管理系统是学校管理机制中重要的一环。图书馆代表着一间学校或者地区的文化标志。因为图书馆丰富的图书资源能够带给我们重要并且优越的学习资源。通过对图书管理系统的运行管理机制进行调查研究，开发了此图书系统。本系统中包含5个功能模块：个人信息，查询图书，读者综合管理，图书综合管理，图书借续还。

关键词：C语言 哈希表 管理系统 控制台界面

目录

[摘要 I](#_Toc28840)

[1. 设计内容和要求 3](#_Toc21586)

[1.1 设计内容 3](#_Toc20228)

[1.2 设计要求 3](#_Toc6355)

[2. 软件需求分析 4](#_Toc19683)

[2.1 功能需求 4](#_Toc32425)

[2.2 系统角色 4](#_Toc14373)

[2.3 系统模块 5](#_Toc13626)

[3. 数据结构设计 6](#_Toc24905)

[4. 系统实现 9](#_Toc10657)

[5. 总结 12](#_Toc14578)

[5.1存在的问题及改进方向 12](#_Toc16163)

[9. 参考文献 13](#_Toc2196)

[10. 致谢 14](#_Toc28383)

# 设计内容和要求

## 设计内容

自己打算实现的功能：

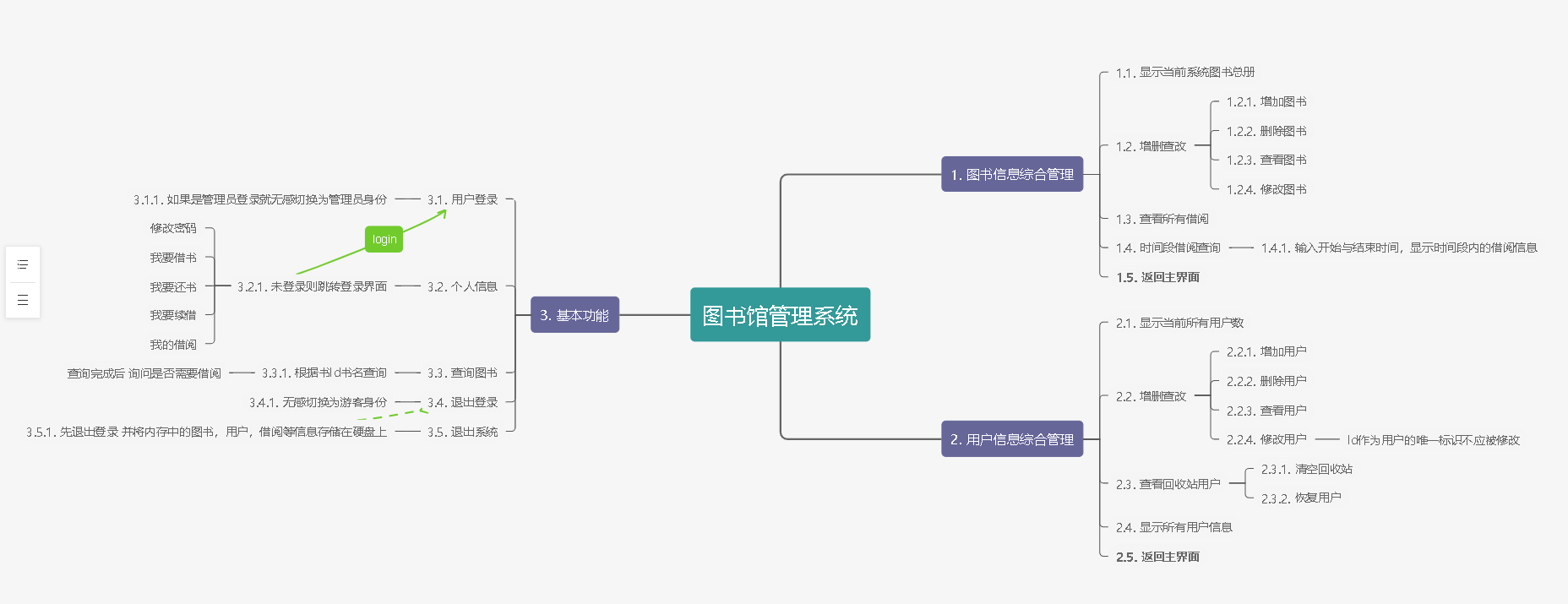
1. 实现读者借阅图书；
2. 实现读者归还图书；
3. 实现对图书信息的综合管理：设置二级子目录实现信息的增，删，改，查等操作。
4. 实现对读者信息的综合管理：设置二级子目录实现信息的增，删，改，查等操作。
5. 显示用户信息：显示所有用户信息，含学号、姓名、借阅状况等信息。
6. 拥有独立的各二级界面，并能够返回主界面。
7. 实现管理员，学生，游客三种身份，且用户身份无感切换。
8. 实现查询时间段内借阅信息。

## 1.2 设计要求

1. 系统至少有图书管理员、用户两种角色。
2. 系统提供控制台界面。
3. 支持从文本文件导入图书信息，导入用户信息。
4. 用二进制文件存储图书信息和用户信息。自定义文件格式。
5. 用文本文件保持日志文件。每天操作保持成一个日志文件。自定义日志文件格式。
6. 图书信息、用户信息从文件读入内存后，保存在哈希表中。
7. 可以增加删除图书，增加删除用户。
8. 可以按书名、作者查询图书；可以按学号、姓名查询用户。
9. 可以查询某一时间段内所有图书借阅情况，例如查询从2022年1月1日到2月13日之间所有被借阅的图书；例如查询从2022年5月9日到6月3日之间所有来借阅图书的同学。
10. 系统分析、设计正确。
11. 界面简洁、美观。

# 软件需求分析

## 2.1 功能需求

图 2-1

## 2.2 系统角色

管理员（admin）拥有最高权限，能显示非管理员用户不能查看的功能：对图书和用户的综合管理二级界面，对图书的信息进行增删查改，对用户信息进行增删查改，支持重置用户密码。并且管理员也应该是用户，也应该能进行正常的借阅操作。

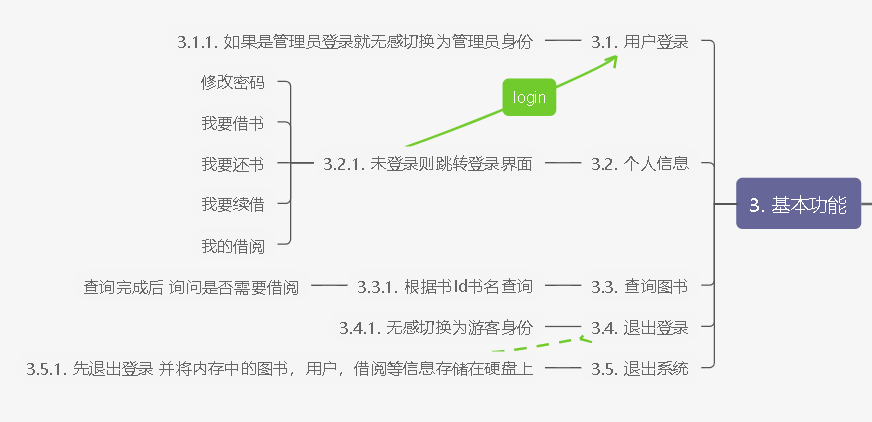
用户（user）作为图书馆管理系统的使用者，能够查询书籍，借续还书，并能查看自己的借阅记录等，支持个人修改密码。

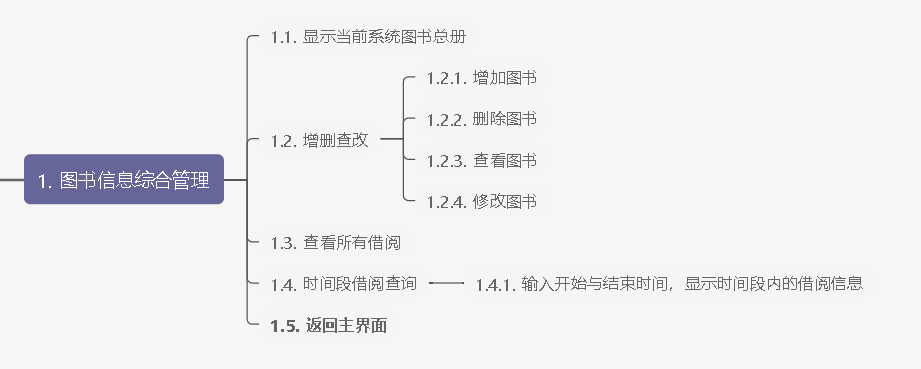
游客（guest）作为系统的默认用户，能够查阅书籍，但如果要进行借阅操作，就需要挑转至用户登录界面进行用户认证。

系统启动时默认以游客身份进行登录，此时系统可以进行查询图书，若进行借阅操作则会跳转到用户登录界面，登录成功后借阅也成功处理。

若使用的是管理员身份进行登录，则主界面会多出2个信息综合管理菜单，这也是管理员对图书和用户进行增删查改的入口，且这两个二级菜单会对当前用户的权限进行判断，如果不是管理员身份则会提示“非法提权”并退出至主界面，权限安全管理。

## 2.3 系统模块

图 2-2基本功能

图 2-3图书信息综合管理

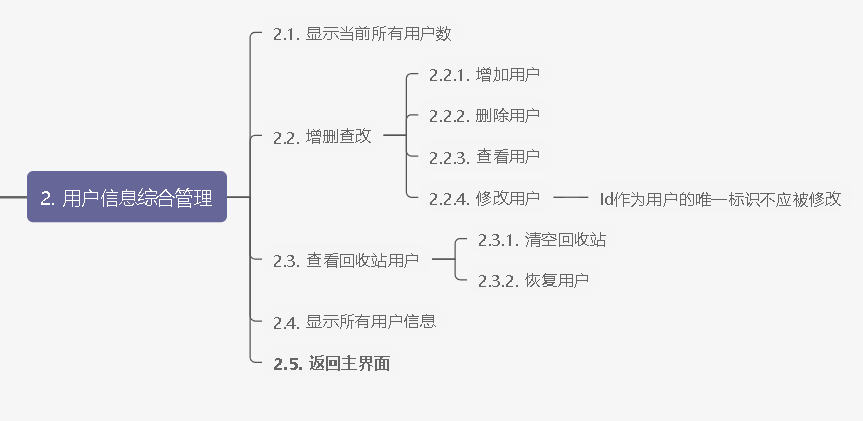


图 2-4用户信息综合管理

# 数据结构设计

user表作为全局变量用于保存当前用户信息，如表3-1所示。

表 3-1user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 是否必填 | 字段说明 |
| 1 | Id | char | 11 | YES | 用户Id |
| 2 | Name | char | 20 | YES | 用户名称 |
| 3 | Password | char | 20 | YES | 用户密码 |
| 4 | Power | int | 1 | YES | 用户权限 |
| 5 | Deleted | int | 1 | YES | 是否标识删除 |

外部有Power为二级指针 内部为字符串指针，

char \*Power[] = {PowerAdmin, PowerUser, PowerGuest};

内部的字符串指针分别指向如下：

char PowerAdmin[] = "Admin";

char PowerUser[] = "User";

char PowerGuest[] = "Guest";

这样即可有单独的索引获得具体的权限信息

比如Power[user.Power]; 返回的就是用户的具体权限

UserLNode表用于保存学生信息,是存放在链表中的节点，如表3-2所示。

表 3-2UserLNode

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 是否必填 | 字段说明 |
| 1 | data | struct UserInfo \* | YES | 用户信息节点，指向链表（即内存）地址 |
| 2 | next | struct UserLNode \* | YES | 下一个节点 |

HashTableNode表用于保存图书信息,是哈希表，如表3-3所示。

表 3-3HashTableNode

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 是否必填 | 字段说明 |
| 1 | TableSize | int | YES | 哈希表数组数量 |
| 2 | HahLists | struct bookIndo \* | YES | 链表头结点 |

BookIndo表用于保存图书详细信息，是哈希表链表中的节点 如表3-4所示。

表 3-4BookIndo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 是否必填 | 字段说明 |
| 1 | Id | char | 11 | YES | 图书Id |
| 2 | Name | char | 600 | YES | 图书名称 |
| 3 | Author | char | 20 | YES | 图书作者 |
| 4 | BookType | int | 1 | YES | 图书种类 |
| 5 | Count | int | 1 | YES | 库存量 |
| 6 | InHandle | int | 1 | YES | 已经被借 |
| 7 | Next | struct bookIndo \* | 1 | YES | 下一个节点 |

RecordInfo表用于保存借阅详细信息，如表3-6所示。

表 3-5RecordInfo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 是否必填 | 字段说明 |
| 1 | BookId | char | 11 | YES | 用户Id |
| 2 | ReaderId | char | 11 | YES | 用户名称 |
| 3 | stime | time\_t | 1 | YES | 用户密码 |
| 4 | etime | time\_t | 1 | YES | 用户权限 |
| 5 | state | int | 1 | YES | 是否标识归还 |

外部有RecordState为二级指针 内部为字符串指针，

char \*RecordState[] = {no, yes, out};

内部的字符串指针分别指向如下：

char no[] = "未归还";

char yes[] = "已归还";

char out[] = "已逾期";

这样即可有单独的索引获得具体的权限信息

比如RecordState[p->state]; 返回的就是该条借阅有没有被归还或者逾期.

# 系统实现

本项目使用vscode进行开发维护，本想使用git进行版本控制，但与父目录的git有所冲突，就未能用上git进行版本控制，可能会对后续的维护带来不便。

系统可以算的上使用了“前后端分离”的思想，前端即用户看见的界面文字等，都声明在show.h实现在show.c中，只进行输入输出与用户交互。所有的后端实现代码也是通过划分模块的方式进行的编写。用户的增删查改等操作实现在user.c中，图书的增删查改实现在bookinfo.c中，借阅记录实现在record.c中，系统运行日志实现在log.c中。

并将各类文件名用#define的方法定义在头文件中，方便后续的更改与生产环境的变化。

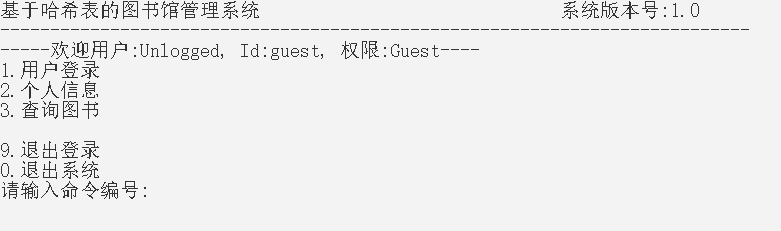


表 4-1默认界面

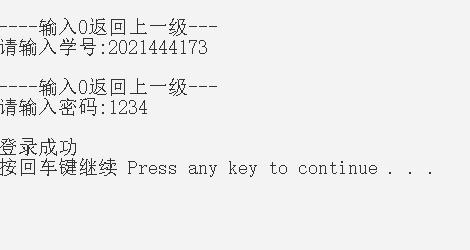


表 4-2登录界面

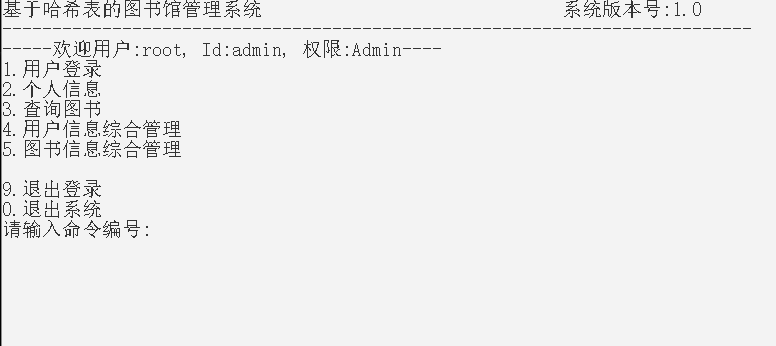


表 4-3管理员用户的主界面

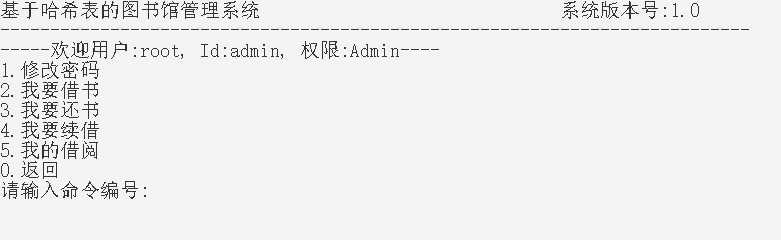


表 4-4个人信息

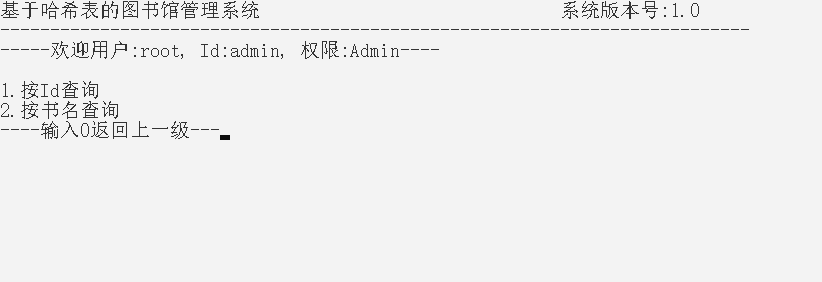


表 4-5查询图书

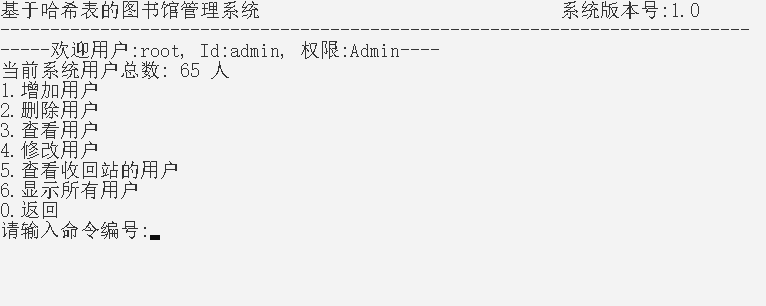


表 4-6用户综合信息

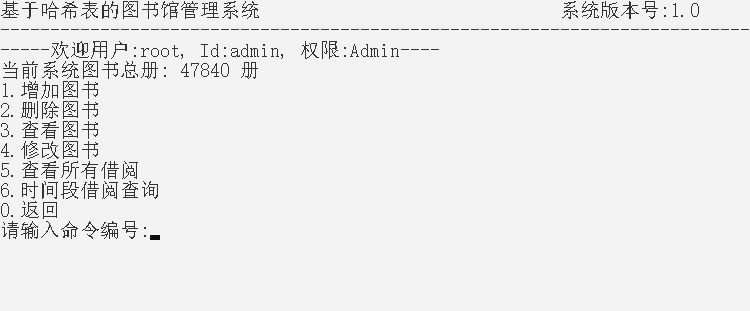


表 4-7图书综合管理

# 总结

## 5.1 项目总结

本项目工程量较大，并不是函数没写好而导致的，而是功能十分的详细，涉及二次确认以及错误操作的响应，这也算是继本人在接触单片机的再一个巨大的工程，本次userinfo.csv使用了自己编写的python脚本进行批量化处理，进行凯撒偏移，与增加power与deleted内容。

## 5.2存在的问题及改进方向

1. 使用的是最简单的凯撒加密，没有设置密文的范围，如果在加密后密文中出现','则会影响csv存储。改进可使用MD5或者SHA1 加密。
2. 在主页面不输入数字，而输入其他的如 cls 会出现主页面的循环，

解决:

getchar();

1. 对于用户信息的存储来说，使用的是单链表存储在内存，读取时使用的是 头插法， 但存储时使用也是头部先保存, 即用户文件在经历一个读入，写出后，顺序会头尾交换。长期如此可能会在硬盘性能有少许影响。
2. 图书和用户信息没有遵循数据的“三二一”原则 不能有效避免灾容、删库、误操作
3. 图书和用户修改的地方，可以加一个是否确认修改的二级询问。
4. log的类别可以分级[INFO DEBUG ERROR WARNING]，可以再记录从什么变为什么， 出现这个问题的原因就是在设计函数时没有多方面考虑。
5. 多个用户同时进行操作时有冲突，解决：借用本人另一个check项目的session方法可解决

# 参考文献

[1] 严蔚敏,李冬梅. 数据结构（C语言版）[M].第2版.北京: 人民邮电出版社,2015.

[2] [美]K.N.King. C语言程序设计现代方法[M]. 吕秀锋,黄倩译.北京: 人民邮电出版社,2021.

[3] 程杰. 大话数据结构[M].北京: 清华大学出版社,2011.

# 致谢

时光飞逝，转眼间，两周的数据结构课程设计即将结束，顺利完成了课程设计任务。首先，我要感谢我们数据结构课程的任课教师\*老师，他的讲解能让我清楚的理解，各类数据结构的特点、差异、以及应用。在进行程序设计时，要注意想好思路。即要有恰当模块名、变量名、常量名、子程序名等。将每个功能的模块，即函数名要清晰的表述出来，使用户能够一目了然此程序的功能。当然适当的给写注释，也是方便用户的理解。还有在编写程序时要注意对程序的适当分配，便于用户看懂程序，也便于自己检查程序。但是完成任何一个较大的程序，都需要掌握一定的编程基础，需要不断的探索和求知过程，这样对自己编程能力的提高有较大的帮助。当然，任何程序必须经过计算机的调试，看是否调试成功，发现错误，一个个，一步步去解决，这样就能从错误中进步。从发现bug到解决bug。

还有各位我请教的和来询问我的朋友们，大佬给了我解决方案，我也解决了各位来着的大部分问题。 值得注意的是我应该多与外界所交流，进行思维的碰撞，不应该单单靠自己的力量。

感谢[microsoft](https://github.com/microsoft)对本项目的支持。

感谢 JetBrains对本项目的支持。