



# LMS简易图书管理系统

## 一.简介

LMS是一个基于 `Linux C` 开发的简易的图书管理系统,拥有用户-管理员的图书管理机制, 人员-图书的操作流程, 并配备了一个基于 `Linux C` 原生终端的可视化模块。

## 二.项目使用(演示)

## 三.项目结构

项目结构可以分为两个板块:用户-图书文件管理系统和 `Linux` 原生GUI

### 1.用户-图书文件管理系统

#### (1).用户类

用户是这个LMS系统中获取信息做出决策的对象,  
按照功能,区分为管理员(可以管理其他用户,管理书籍等等)和普通用户

```

//
// Created by 29396 on 24-7-13.
//

typedef struct borrow {
    ull id;           //书籍ID
    ull date;         //书籍到期时间
} borrow;

void borrow_copy(borrow& to, borrow& from);

typedef struct User {
    ull ID;           //用户ID
    char name[52];     //昵称
    char password[52]; //密码
    borrow Borrow[5];  //借的书籍
    bool power;        //0为读者, 1为管理员
} User;

//拷贝函数
void User_copy(User& to, User& from);

//返回用户数量 0
int get_user_num();

//添加用户 0
int append_user(User* add_user);

//删除用户 0
int del_user(ull del_ID);

//给定 ID和密码进行比对, 匹配成功返回0, 失败返回1, 没有该用户返回2, 出现错误返回-1 0
int pwd_comparison(ull ID, char* pwd);

//给定ID, 获得对应用户结构体 0
User* get_user(int ID);

```

```
//给定更新的用户结构体，更新成功返回0； 0  
int user_update(User* update_user);  
  
//返回用户的数组指针，page表示加载到第几页 0  
User* list_user(int page);
```

## (2).图书类

按照需求,部分图书对普通用户不可见

```

//记录图书存放位置
typedef struct where {
    int floor;           //楼层
    char zone;           //区域
    int shelf;           //书架
    int number;          //编号
} where;

//where的拷贝函数
void where_copy(where &to, where &from);

typedef struct Book {
    int id;              //书籍ID
    char ISBN[15];       //ISBN码
    char Title[52];      //标题
    char tag[11];        //标签
    where Where;          //书籍位置
    bool visual;         //对普通读者是否可见
    ull status;          //在馆内为0，借出记录为用户ID
} Book;

void book_copy(Book &to, Book &from);

// 返回图书数量
int get_book_num();

//added_book 被添加的图书指针，返回操作状态码。
int add_book(Book *added_book);

//del_id 被删除书籍的id，返回操作状态码
int del_book(ull del_id);

//sk_id 需要查询的书籍的id，返回id匹配的书籍
Book* seek_book_by_id(ull sk_id);

//update_book 被更新书籍的指针，返回操作状态码
int update_book(Book *update_book);

//返回书籍列表指针

```

```
Book* list_books();
```

### (3).标签类

标签类是每一本图书对应的标签

```
class Tag{
public:
    Tag();

    ~Tag();

    int Write_tag();

    int Update_tag(char Tag[]);

    int Show_tags();

    int get_tag_num();

    char* seek_tag(int index);

private:
    int tag_num;
    char tag_list[][50];
};
```

### (3)Display类

display类是主体

通过调用自己BOOK和USER子类的函数实现对用户和书籍的操作  
并调用自身函数实现可视化

```

//
// Created by lilon on 24-7-10.
//

class Display {
public:
    void LogRes(); //登录&注册
    int mainpage(); //主界面

    int check(); //查看所有书籍

    int manage(); //书籍管理

    int search(); //查询&排序
    void seekID_paint(bool if_show);
    void seekTitle_paint(bool if_show);
    void seekID();
    void seekTitle();

    int usrman(); //用户管理
    int lbook(); //借还书
    int modify(); //查询&修改个人信息
    void loading(); //载入界面

protected:
    Book book{};
    User user{};
    //页面栈(开发中)
    /*int head=0,tail=0;
    int page[10]={0};*/

private:
};

```

## (4)main函数

```
int main() {
    Display display;
    START:
    display.LogRes();
    while(true) {
        int pagecode=display.mainpage();
        switch (pagecode) {
            case 0:
                display.LogRes();
                break;
            //Book Check
            case 1:
                display.check();
                break;
            //Book Management
            case 2:
                display.manage();
                break;
            //Search
            case 3:
                display.search();
                break;
            //User Management
            case 4:
                display.usrman();
                break;
            //Mine
            case 5:
                display.modify();
                break;
            default:
                display.lbook();
                break;
        }
    }
}
```

## 四.项目亮点

本项目的亮点是不依赖于额外库的，使用原生终端实现了一个简易的可视化图形界面  
(依赖演示)