

LibrarySystem使用说明

该图书管理系统需要使用SQLite数据库存储对应的信息，所以在安装LibrarySystem（以下称为图书管理系统）时需要安装SQLite数据库。

构建说明

在项目文件夹下有一个名为 `build.sh` 的文件，这是构建图书管理系统的脚本文件。在你执行该脚本文件的时候，首先会检查你的计算机上是否安装 `SQLite3` 和 `CMake` 程序，如果你的计算机上没有这两个程序，那么 `build.sh` 会提示你按步骤进行安装。只有 `SQLite3` 和 `CMake` 程序都在你的计算机上安装时，才能执行编译步骤。

该脚本支持Windows、Linux、Unix操作系统，执行构建脚本的命令：

```
./build.sh
```

在你成功执行 `build.sh` 脚本后，编译后的程序会放在当下目录下，其默认的名称为 `LibrarySystem`。如果你不喜欢这个名字，你可以通过修改 `CMakeLists.txt` 文件修改名字：

```
cmake_minimum_required(VERSION 3.10)
project(LibrarySystem)

set(CMAKE_CXX_STANDARD 20)
set(CMAKE_CXX_STANDARD_REQUIRED ON)

find_package(SQLite3 REQUIRED)

add_executable(LibrarySystem src/main.cpp src/database.cpp src/sha256.cpp
)

target_include_directories(LibrarySystem PRIVATE ${CMAKE_CURRENT_SOURCE_DIR})

target_link_libraries(LibrarySystem PRIVATE SQLite::SQLite3)
```

在 `CMakeLists.txt` 文件中，`add_executable(LibrarySystem ...)`，将这里的 `LibrarySystem` 替换成你想要输出的程序名称。

还有一种办法就是重命名编译后的文件。

项目结构

```
LibrarySystem-----项目文件夹
|
| header-----头文件
| | -----database.h-----数据库操作接口
| | -----sha256.h-----数据库信息加密算法
| | -----utils.h-----终端UI接口
| src-----源文件
| | -----database.cpp----数据库操作接口实现源文件
| | -----sha256.cpp----加密算法实现源文件
| | -----main.cpp-----主程序猿文件
build.h ----- 项目构建文件
CMakeLists.txt ----- CMake
```

本项目所有函数的命名和变量的命令均由中文经过微软翻译得到。

LibrarySystem使用说明

LibrarySystem（图书管理系统）在登录界面提供了管理员登录和学生登录，两者共用同一套系统。在系统初次启动时会在程序所在的当前文件夹下创建一个名为 **Library.db** 的数据库文件，所有的图书信息和用户（包括管理员、学生）的信息都会存在这个数据库内。用户的数据会通过简单的SHA256算法进行加密（难的不会写，OpenSSL装起来又麻烦，如果想要使用更复杂的加密，可以自己安装OpenSSL并调用相关API进行加密操作）。

本系统可实现管理员用户对图书信息的增删改查功能，也能看到学生的借阅信息和借阅状态。除此之外，可以修改学生的密码。

```
管理员初始账户：admin
管理员初开密码：admin
```

管理员用户可以添加管理员用户，学生用户在创建的时候需要填写基本信息，这些信息作为借阅书籍时的登记信息。