

# 书店管理系统

SJTU ACM班 2021级 程序设计课程大作业

## 目录

-  [简介](#)
-  [作业说明](#)
  -  [评测方式](#)
  -  [分数组成](#)
  -  [中期检查](#)
  -  [检查时间表](#)
-  [项目说明](#)
  -  [开发要求](#)
    - [开发文档](#)
    - [Github 仓库管理](#)
    - [代码风格](#)
  -  [业务需求](#)
    - [基本要求](#)
    - [交互方式](#)
    - [账户系统](#)
    - [图书系统](#)
    - [日志系统](#)
    - [性能需求](#)
  -  [Bonus](#)

## 简介

- 实现一个用 C++ 语言编写的书店管理系统，用于向店家和顾客提供图书检索与购买相关服务
- 本项目目标如下
  - 培养学生工程开发的能力（开发文档的编写与程序的封装等）
  - 培养学生的代码管理能力（Github 的使用），拥有良好的代码规范意识
  - 提升学生的编程学习与实践能力（块状链表的学习实现）

- 培养学生对编写软件的兴趣（见 [Bonus](#)）
- 关于本仓库
  - 本仓库为模板仓库，可以点击上方绿色的 `Use this template` 按钮复制出一份属于你的仓库用于完成作业。助教也十分鼓励学生自行新建仓库完成作业
    - 本仓库中主要文件为助教提供的块状链表链接库，具体使用方式见仓库中 `bin` 和 `src` 文件夹中内容
    - 如果使用仓库模板，应该自定义 `README.md` 内容（其中如 这样的 Shield，将链接中的 `/:user/:repo` 改成个人信息即可，详见 [Shields.IO](#)）
  - 关于本仓库有任何问题欢迎联系助教，也可以直接 Pull request
  - 本仓库内容将维护至本作业结束，请保持关注页面更改，助教也会通过 QQ 群发布相关通知



## 作业说明



## 评测方式

- 公开的评测数据见本仓库下 `asset/Data.rar`
  - 注意：每个 `testcase` 为最小评测数据单元。即测试同一 `testcase` 的多个输入文件过程中，对于每个输入文件，都会以其为输入运行学生程序，完成后关闭程序，再以下一个输入文件运行学生程序，但过程中不会清除学生程序运行生成的数据文件。而每个 `testcase` 测试结束后会清除所有学生程序生成的文件，再运行下一个 `testcase`。
- Online Judge（以下简称 OJ）提交方式为提交 `cpp` 或 `Git` 格式的代码
  - 若提交格式为 `Git`，则提交内容应为形如 `https://github.com/username/repo.git` 的链接
    - 编译评测方式详见 OJ 首页说明。注意：编译生成的可执行文件名应为 `code`；OJ 使用沙盒评测文件读写
    - 由于网络原因可能会导致评测机 Clone 仓库失败。仓库内容过大（如将解压后的数据文件或 Build 内容一并放入仓库）导致的问题后果自负。（Tip：可以使用 Github 仓库加速通道链接提交作业，例如 Chrome 插件 [GitHub加速](#)）
- 本地评测可用脚本评测，参考 `asset/localJudge.py`
- 评测数据量（账户和图书数量）不会超过  $10^5$  数量级

## 分数组成

- 本作业满分为 100%，最终将以本作业占本课程成绩总分分数折算入课程成绩
- 注意：使用助教提供的链接库的 OJ 提交得分以正常得分的 60% 计算

得分项	分值	说明
正确性	55%	通过所有基础数据点 (得分计算方式为： $\text{OJ历史提交最高得分} / 100 * 55\%$ )
鲁棒性	10%	通过所有里数据点
开发文档	10%	助教评分 5% + 小组评分 5%
Code Reivew	15%	Github 仓库管理、代码规范等
Bonus	10%	具体各项得分见 <a href="#">Bonus</a>

## 中期检查

- 助教将在固定时间节点检查学生的完成进度，具体时间详见 [检查时间表](#)
  -  请所有学生填写问卷：[Bookstore-2021 学生信息统计](#)
- 中期检查将以仓库的 Tag 为准，助教将检查最新 Tag 对应的 Commit 的内容是否符合该阶段要求
- 如遇困难请提前联系助教

## 检查时间表

- **Week 0** (对应校历 2021-2022 学年第一学期第 12 周)
  - 周一 11.29：发布项目、创建 Github 仓库
- **Week 1**
  - 周一 12.6：提交、检查、交换[开发文档](#)
- **Week 2**
  - 周一 12.13：检查文件存储数据结构（块状链表）进度
  - 周五 12.17：通过文件存储数据结构正确性评测 ([OJ 链接\(P1412\)](#))
- **Week 3**
  - 周三 12.22：检查主体逻辑部分进度
- **Week 4** (对应校历第 16 周)
  - 周一 12.27：检查[主体逻辑部分正确性测试](#)
    - 通过测试者进行 Code Review
  - 周五 12.31：通过主体逻辑部分正确性测试
    - OJ 测试由两部分组成：[公开数据与基础鲁棒性评测 \(P1075\)](#)；[里数据评测 \(P1411\)](#)
    - 所有学生完成 Code Review

## ■ Bonus 成果展示

友情提醒：期末考试总是要好好备考的，所以请务必不要留给第 16 周的自己太多工作量🙏

## 项目说明

---

### 开发要求

#### 开发文档

- 每位学生均需完成一份开发文档，且该文档将由另一位学生负责实现
  - 文档交换遵循自愿原则，如学生无法自行达成交换共识，则由助教指定交换方式
- 开发文档格式无硬性要求
  - 排版软件可以使用 Markdown, Latex, Word 等
- 开发文档内容**无硬性要求**，内容可以参考以下条目
  - 项目名称 / 文档作者
  - 程序功能概述
  - 主体逻辑说明
  - 代码文件结构，即 `main` 函数与各个类的关系
  - 各个类的接口及成员说明
  - 具体算法说明
  - 其他补充说明
- 如果最后提交代码不符合开发文档则会对实现者**严重扣分**。合理的文档修订是允许的，但请妥善保存任何文档修改情况及对应的文档编写者与实现者的交流记录

#### Github 仓库管理

- 了解版本库、工作区、暂存区等 Git 基础概念
- 掌握 `commit` 等常用指令（可参考 Walotta 助教于 2021.11.1 晚的上机课授课内容）
- Github 仓库整体文件结构清晰，了解并自主设置 `README.md`, `.gitignore`, `LICENSE`, `.github/` 等文件
- 合理使用 Commit 和 Tag 功能维护代码版本
  - Commit Message 内容可以清晰简要但不能缺失
  - Tag 命名合理，可参考 [Wikipedia: Software versioning](https://en.wikipedia.org/wiki/Software_versioning)
- 额外要求
  - 学习 Branch 的使用
  - 学习 Github Action 的使用

## 代码风格

- 选择合适的代码风格，并在仓库说明文档中说明
- 严格遵守选定风格的代码规范
- 额外要求
  - 学习动态链接库的生成与使用

## 业务需求

### 基本要求

- 设计并实现一个书店管理系统，最终交接物为开发文档、工程源代码及可执行文件
- 关于程序有以下要求
  - 编程语言限制：软件主体完全使用 C++ 编写
  - 程序首次运行时自主执行所需的初始化操作
    - 创建用户名为 `root`，密码为 `sjtu`，权限为 `{7}` 的超级管理员账户
  - 程序启动后，根据输入数据执行对应操作，直至读入 EOF 或根据相关指令要求终止运行

### 交互方式

- 支持使用命令行（Windows CMD 或 Linux Terminal）进行操作，输入数据以换行符或回车符为分隔为若干指令。本文档中以弱化的正则语法说明指令文法。合法指令的文法相关说明如下，更多指令具体说明见下文
  - 指令合法字符集为标准 Ascii 字符
  - 单个指令最大长度为 `1024` 字节
  - 允许的空白符仅为空格，单个指令被空白符分割为多个部分
    - 多个连续空白符效果与一个空白符等价
    - 行首行末允许出现多余空白符
    - 如无特殊说明禁止省略或另增空白符
  - 指令中第一个部分必须为指令关键词，指令中关键词部分必须与指令格式完全匹配
  - `[x]` 表示一串有特定限制的用户自定义字符串
  - `(a|b|c)` 表示此处仅能出现 `a`, `b`, `c` 中其一
  - `(x)?` 表示此处可以出现零次或一次 `x`
  - `(x)+` 表示此处可以出现一次或多次 `x`

```
# 基础指令
quit
exit
```

```

# 账户系统指令
su [User-ID] ([Password])?
logout
register [User-ID] [Password] [User-Name]
passwd [User-ID] ([Old-Password])? [New-Password]
useradd [User-ID] [Password] [Priority] [User-Name]
delete [User-ID]

# 图书系统指令
show (-ISBN=[ISBN] | -name="[Book-Name]" | -author="[Author]" | -keyword="[Keyword]")?
buy [ISBN] [Quantity]
select [ISBN]
modify (-ISBN=[ISBN] | -name="[Book-Name]" | -author="[Author]" | -keyword="[Keyword]" | -price=[Price])+
import [Quantity] [Total-Cost]

# 日志系统指令
report myself
show finance ([Time])?
report finance
report employee
log

```

- 在用户输入一条指令后，如果合法则执行对应操作，如果有则输出操作结果；如果指令非法或操作失败则输出 `Invalid\n`
  - 仅有空白符的指令合法且无输出内容
  - 除非有特殊说明，如果输入指令对应的输出内容非空，则结尾有 `\n` 字符；输出内容为空则不输出任何字符
- `quit` 和 `exit` 指令功能为正常退出系统

## 账户系统

- 书店管理系统服务于 **店主（超级管理员）**，**员工（管理员）** 以及 **顾客（用户）**。为满足其不同需求，故需账户系统管理账户及权限数据，并提供相应功能
- 权限说明
  - 本文档中以 `{x}` 表示权限等级，数字越大则权限越高
  - 账户的权限等级有 `{7}`，`{3}` 和 `{1}`
  - 未登录任何账户的状态下当前账户权限视为 `{0}`
  - 低权限账户能够访问的功能严格是高权限账户能够访问的功能的 **子集**
  - 指令的权限等级表示能够执行该指令的最低当前账户权限需求
- 各权限可访问功能说明
  - **店主 {7}**：使用 `root` 账户，可以访问书店系统所有功能，包括日志和账户管理功能
  - **员工 {3}**：可以访问修改图书数据相关功能等
  - **顾客 {1}**：可以注册账户，查询 / 购买图书
  - **游客 {0}**：可以注册账户

- 本系统中有且仅有以上四种权限等级，其余权限等级非法
- 登录栈
  - 系统支持嵌套登录，即允许复数个账户同时处于登录状态
- 输入的指令视为最后登录的账户操作
- 退出系统视为登出所有已登录账户
- 账户信息说明
  - 关于指令格式中的用户自定义字符串限定如下，其中“最大长度”含义为该字符串所占存储空间不能超过指定数目的英文字母字符所占存储空间
  - `[User-ID]`, `[Password]`, `[Old-Password]`, `[New-Password]`
    - 合法字符集：数字，字母，下划线
    - 最大长度：30
  - `[User-Name]`
    - 合法字符集：除不可见字符以外 Ascii 字符
    - 最大长度：30
  - `[Priority]`
    - 合法字符集：数字
    - 最大长度：1
  - 出现不符合上述限定的自定义字符串的指令视为非法，下同
- 指令说明
- 本文档中指令说明格式如下
  - 指令名称
    - `{指令权限} [指令格式]`
    - 指令功能
      - 指令逻辑（条数不定，各逻辑间优先顺序同列举顺序）
  - 登录账户指令
    - `{0} su [User-ID] ([Password])?`
    - 使登录账户变为已登录状态，当前账户变为该账户
      - 如果密码错误则操作失败
      - 如果当前账户权限等级高于登录账户则可以省略密码
  - 注销账户指令
    - `{1} logout`
    - 撤销最后一次成功执行的 `su` 指令效果
      - 如无已登录账户则操作失败
  - 注册账户指令
    - `{0} register [User-ID] [Password] [User-Name]`
    - 注册信息如指令格式所示，权限等级为 `{1}` 的账户
      - 如果 `[User-ID]` 与已注册账户重名则操作失败
  - 修改密码指令

- `{1} passwd [User-ID] ([Old-Password])? [New-Password]`
- 修改指定账户的密码
  - 如果该账户不存在则操作失败
  - 如果密码错误则操作失败
  - 如果当前账户权限等级为 `{7}` 则可以省略 `[Old-Password]`
- 创建账户指令
  - `{3} useradd [User-ID] [Password] [Priority] [User-Name]`
  - 创建信息如指令格式所示的账户
    - 如果待创建账户的权限等级大于等于当前账户权限等级则操作失败
    - 如果 `[User-ID]` 与已注册账户重名则操作失败
- 删除账户指令
  - `{7} delete [User-ID]`
  - 删除指定账户
    - 如果待删除账户不存在则操作失败
    - 如果待删除账户已登录则操作失败

## 图书系统

- 图书系统提供图书信息、库存信息和图书交易财务记录的相关服务及数据存储功能
- 图书信息说明
  - `[ISBN]`：Internal Strange Book Number
    - 合法字符集：除不可见字符以外 Ascii 字符
    - 最大长度：20
    - 特殊说明：本系统中任何两本图书不应有相同 `[ISBN]` 信息
  - `[Book-Name]`, `[Author]`：图书名字，图书作者名字
    - 合法字符集：除不可见字符和英文双引号以外 Ascii 字符
    - 最大长度：60
  - `[Keyword]`：图书关键词
    - 合法字符集：除不可见字符和英文双引号以外 Ascii 字符
    - 最大长度：60
    - 特殊说明：`[keyword]` 内容以 `|` 为分隔可以出现多段信息。如 `math|magic|quantum` 表示该图书具有三个无顺序的关键词 `math`, `magic` 和 `quantum`
  - `[Quantity]`：购买数量
    - 合法字符集：数字
    - 最大长度：10
    - 特殊说明：数值不超过 `2'147'483'647`
  - `[Price]`, `[Total-Cost]`：图书单价，交易总额
    - 合法字符集：数字和 `.`
    - 最大长度：13
    - 特殊说明：本系统中浮点数输入输出精度固定为小数点后两位
- 字符相关信息默认值为空，数字相关信息默认值为数值 `0`



- 指令说明

- 检索图书指令

- `{1} show (-ISBN=[ISBN] | -name="[Book-Name]" | -author="[Author]" | -keyword="[Keyword])?`
- 以 `[ISBN]` 字典升序依次输出满足要求的图书信息，每个图书信息输出格式为 `[ISBN]\t[Book-Name]\t[Author]\t[Keyword]\t[Price]\t[库存数量]\n`
  - 无满足要求的图书时输出空行
  - 无附加参数时，所有图书均满足要求
  - 附加参数内容为空则操作失败
  - `[Keyword]` 中出现复数个关键词则操作失败

- 购买图书指令

- `{1} buy [ISBN] [Quantity]`
- 购买指定数量的指定图书（由于本系统不包括支付系统，故仅需减少库存）
  - 没有符合条件的图书则操作失败

- 选择图书指令

- `{3} select [ISBN]`
- 以当前账户选中指定图书（账户登出后无需保存选中图书情况）
  - 没有符合条件的图书则创建仅拥有 `[ISBN]` 信息的新图书
  - 退出系统视为取消选中图书

- 修改图书信息指令

- `{3} modify (-ISBN=[ISBN] | -name="[Book-Name]" | -author="[Author]" | -keyword="[Keyword]" | -price=[Price])+`
- 以指令中的信息更新选中图书的信息
  - 如未选中图书则操作失败
  - 有重复附加参数为非法指令
  - 附加参数内容为空则操作失败
  - `[keyword]` 包含重复信息段则操作失败

- 图书进货指令






- `{3} import [Quantity] [Total-Cost]`
- 以指定交易总额购入指定数量的选中图书，增加其库存数
  - 如未选中图书则操作失败

## 日志系统

- 日志系统负责记录书店管理系统运行过程中的各种数据，提供各类日志信息查询服务

- 日志信息说明

- `[Time]`：交易笔数
    - 合法字符集：数字
    - 最大长度：10

- 特殊说明：数值不超过 `2'147'483'647`
- 指令说明
  - 生成员工操作记录指令
    - `{3} report myself` 
    - 返回员工自己的操作记录，格式自定
  - 财务记录查询指令
    - `{7} show finance ([Time])?`
    - 输出最后完成的指定笔数交易总额，格式为 `+ [收入] - [支出]\n`，例如 `+ 1.23 - 45678.90`
      - 无 `Time` 参数时，输出所有交易之总额
        - `Time` 为 0 时输出空行
        - `Time` 大于历史交易总笔数时操作失败
  - 生成财务记录报告指令
    - `{7} report finance` 
    - 生成一张赏心悦目的财务报表，格式自定
  - 生成全体员工工作情况报告指令
    - `{7} report employee` 
    - 生成一张赏心悦目的员工工作情况表，记录其操作，格式自定
  - 生成日志指令
    - `{7} log` 
    - 返回赏心悦目的日志记录，包括系统操作类的谁干了什么，以及财务上每一笔交易情况，格式自定
- 注意：带有  标记的指令不会出现在测试数据中，但需要实现并手动向助教展示。（被标记的指令的说明来自于祖传题面）

## 性能需求

- 考虑到应用场景下用户和图书数量较大，故禁止将主体数据存储在内存，应实时读写文件数据（可以简单理解为 `exit` 指令的效果等价于强行终止程序）
  - 禁止类似 File As Database 的实现方式（如果你不知道这是什么意思，那么也不用担心自己会触犯该条要求）

## Bonus

 反卷从我做起，Bonus 是用来进一步学习的，有轻松拿满的路线，也有屠龙难度的路线

- 标记 `<k>` 表示得分。各任务说明中得分标记表示该任务得分满分
- Bonus 部分总满分 `<10>`，得分溢出则以 `<10>` 计
- 除 Task 18 外，所有任务均无正确性测试，由学生展示后助教主观评分
- Task 4~16 若展示成果极为优秀，可视情况给分溢出该任务满分
- 有意愿尝试 Task 14~18 的学生请联系助教

## 满减套餐

- **Task 1.** `<1>` 美观的 Report
  - 以你的审美实现 `report` 指令输出一份美观的报告
- **Task 2.** `<1>` 数字选择的 CLI 界面
  - 即各种脚本提供的界面。如下例

```
欢迎光临, PaperL. 您需要哪种服务? :
```

```
1. 买书
2. 卖二手书
3. 砸书店
> 2
```

```
请输入 "<书名> <数量>":
> 《他改变了电院》 3337
```

```
交易失败😂本店不收该书
按任意键返回菜单
>
```

- 请在 CLI 界面中使用不同颜色的文本帮助用户更好地阅读界面
- **Task 3.** `<1>` 支持中文 & emoji 显示
  - 输入数据应支持中文
  - 输出数据应支持中文和 emoji（即可在界面中使用 emoji）

## 工业黑盒

- 加密存储
- 尝试在数据被窃取后保证数据安全，在以下方案中择一实现（同时完成 Task 4, 5 仅得 `<3>`）
  - **Task 4.** `<2>` 支持设置密保问题
  - **Task 5.** `<3>` 通过本地机器特征码加密文件中的数据
- **Task 6.** `<2>` 将自己的文件存储数据结构封装为动态链接库，并在头文件中加注清晰明确的注释
  - **Task 7.** `<1>` 在完成 Task 6. 的基础上，制作一个书店管理系统安装包，实现更合理地放置程序本体和数据文件，并向用户提供软件快捷方式的效果

## 自动化

- 尝试使用 Github Action 对上传代码自动执行指定操作。在以下两个功能中实现一个得 `<3>`，实现两个得 `<4>`
  - **Task 8.** 进行 Lint 检查
  - **Task 9.** 编译生成并 Release 多平台可执行文件
- **Task 10.** `<6>` 自动以公开数据评测程序正确性，并生成报告

## 安全第一

- 支持自动定时备份数据，并可一键将数据恢复至指定备份版本，在以下方案中择一实现
  - **Task 11.** <3> 完全备份
  - **Task 12.** <7> 差异备份
  - **Task 13.** <9> 增量备份

## 屠龙神技

- **Task 14.** <8> 语法高亮（Posix 平台）
- **Task 15.** <8> Tab 补全（Posix 平台）
- **Task 16.** <10> 高并发块链
  - 支持对于特殊数据的多线程读取
    - 该数据仅有读取操作，允许程序离线读入所有指令后，并发进行文件读取行为
- **Task 17.** <10> GUI 前端
- 实现网页或 Python 等 GUI 界面
- **Task 18.** <10> B+ 树
  - 以 B+ 树而非块状链表实现本作业文件存储功能
  - 该功能需要通过 OJ 正确性与时间复杂度评测