

bookstore 开发文档

ACM2022- 金嘉禾

程序功能概述

交互

用户使用命令行进行读入和输出，系统进行检测。若合法则执行对应操作，若指令非法或操作失败则报错 Invalid.

账户管理

每个账户包含信息：username,userId,privilege,password.

账户可执行指令：登录账户、退出账户、注册账户、修改密码、创建账户、删除账户。

图书管理

每种图书包含信息：ISBN,bookname,author,keyword,quantity,price.

图书可执行指令：

show: 基于 ISBN/书名/作者名/关键词检索，以 ISBN 字典升序依次输出满足要求的图书信息。

buy: 指定ISBN与quantity进行购买。

select: 使当前用户选中某本图书。

modify: 以指令中的信息修改选中图书的信息。

import: 以指定交易总额购入指定数量的选中图书，增加其库存数。

日志管理

日志包含信息：交易笔数 transcation_num, 操作次数 record_num, 总结余 total, 总收入 earn_total, 总支出 cost_total.

可实现指令：

show finance: 查询最近指定笔数的财务记录（包含支出与收入）。

log: 展示书店的营业日志。

主题逻辑说明

- 在main函数中执行一个循环，读入每一行指令command（包含一个try-catch语句来进行错误处理）。
- 使用ProcessLine函数对指令进行切片，并由第一条切片判断指令种类，将切片后的指令传给相应的操作函数。
- 操作函数调用对应类的成员函数，进行操作。
- 调用各个类的成员函数执行操作，使用文件读写，从每个类所对应的文件中入读数据，并将修改后的数据写入文件中。
- 在上述的任何一个环节。如果出现错误或指令无法完成，均抛出异常，由main 函数捕获，输出 Invalid.

代码文件结构

文件：

主程序：main.cpp

切分指令并调用处理函数文件组：bookstore.cpp bookstore.h

账户相关文件组：account.cpp account.h

图书相关文件组：book.cpp book.h

登录状态相关文件组：status.cpp status.h

日志相关文件组：log.cpp log.h

字符串处理文件组：strlib.cpp strlib.h

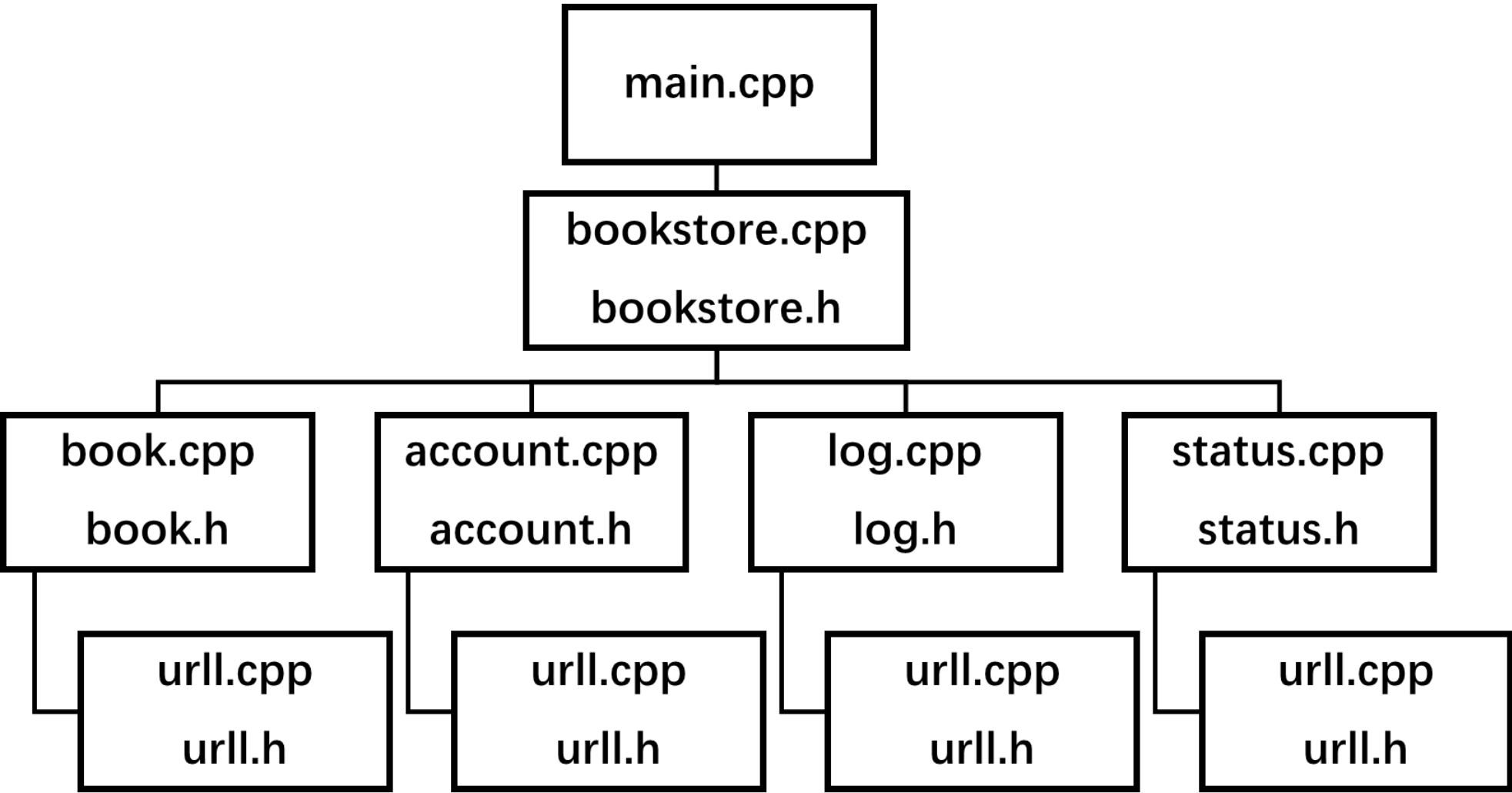
文档存储处理文件组：urll.cpp urll.h

类：

- 账户信息类：AccountFile
- 图书信息类：BookFile
- 登录状态类：LogStatus
- 块状链表类：UnrollLink

函数：

- main函数
- 切分解析输入指令函数: ProcessLine
- 指令执行函数：su, RegisterUser, passwd等
- 账户信息类成员函数
- 图书类成员函数
- 日志类成员函数
- 登录状态类成员函数
- 块状链表类成员函数



各类接口文件说明

account.h

AccountFile 类：

数据成员：

- `id_num`: 一个块状链表类对象，以 `id` 为关键字，以用户信息在文件中的存储位置为值
成员函数：
- `RegisterAccount`: 根据 `UserId,password,name` 注册一个权限为1的账户。在文件file/account_file 中写入信息，在 `id_num`中插入 `Userid` 与当前文件读写指针的位置 `num`.
- `ChangePassword`: 修改 `ID=UserId` 用户的密码为 `new_password`.
- `CreatAccount`: 根据 `UserId,password,name,priority` 注册账户。在文件 file/account_file 中写入信息，在`id_num` 中插入 `Userid` 与当前文件读写指针的位置 `num`.

- `DeleteAccount`: 删除 `ID=UserId` 的用户。

```
class AccountFile{
public:
    UnrollLink id_num;
    AccountFile();
    ~AccountFile();
    void RegisterAccount(const char *UserId,const char *password,const char *name,TransactionLog&);
    void ChangePassword(const char *UserId, const char *new_password, LogStatus
&log_status,TransactionLog&,const char *current_password="");
    void CreatAccount(const char *UserId,const char *password,const int priority,const char *name,LogStatus
&,TransactionLog&);
    void DeleteAccount(const char *UserId,LogStatus &,TransactionLog&);
    std::fstream ioF;
};
```

book.h

BookFile 类:

数据成员:

- `isbn_num`: 一个块状链表类对象, 以 `ISBN` 为关键字, 以图书信息在文件中的存储位置 `num` 为值。
- `bookname_num`: 一个块状链表类对象, 以 `bookname` 为关键字, 以用户信息在文件中的存储位置 `num` 为值。
- `author_num`: 一个块状链表类对象, 以 `author` 为关键字, 以用户信息在文件中的存储位置 `num` 为值。
- `keyword_num`: 一个块状链表类对象, 以 `keyword` 为关键字, 以用户信息在文件中的存储位置 `num` 为值。

成员函数:

- `show_ISBN`: 展现指定 `ISBN` 的图书。
- `show_name`: 展现指定 `name` 的图书。
- `show_author`: 展现指定 `author` 的图书。
- `show_keyword`: 展现指定 `keyword` 的图书(只能有一个关键词, 否则为非法输入)。
- `show_all`: 展现书店内所有图书。
(展现: 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[BookName]\t[Author]\t[Keyword]\t[Price]\t[Quantity]\n`, 其中 `[Keyword]` 中关键词顺序为输入时的顺序。)
- `buy`: 购买数量为 `quantity` 的选中图书。
- `TryModifyx`: 尝试修改选中图书的参数 `x` (`x=ISBN/name/author/keyword`)。如果检测到非法输入, 则抛出异常。如果操作合法, 不做修改。
- `Modifyx`: 修改选中图书的参数 `x` (`x=ISBN/name/author/keyword`)。
(这样操作是为了防止当一条多参数Modify信息的后部参数出错时也对前部参数进行了修改。正确的操作是抛出异常, 不做任何修改。)
- `import`: 书店进货数量为 `quantity` 的选中图书。

```
class BookFile{
public:
    BookFile();
    ~BookFile();
    void show_ISBN(const char *ISBN);
    void show_name(const char *name);
    void show_author(const char *author);
    void show_keyword(const char *keyword);
    void show_all();
    void buy(const char *ISBN,const int &quantity,TransactionLog &,LogStatus&,AccountFile&);
```

```

void TryModifyISBN(const char *ISBN,LogStatus& );
void TryModifyname(const char *name,LogStatus&);
void TryModifyauthor(const char *author,LogStatus&);
void TryModifykeyword(const char *keyword,LogStatus&);
void TryModifyprice( long long price,LogStatus&);
void ModifyISBN(const char *ISBN,LogStatus& );
void Modifyname(const char *name,LogStatus&);
void Modifyauthor(const char *author,LogStatus&);
void Modifykeyword(const char *keyword,LogStatus&);
void Modifyprice( long long price,LogStatus&);
void import(int quantity,long long cost,LogStatus&,TransactionLog &,LogStatus&,AccountFile&);
UnrollLink isbn_num ;
UnrollLink bookname_num;
UnrollLink author_num;
UnrollLink keyword_num;
std::fstream iof;
};

```

status.h

LogStatus 类:

数据成员:

`std::vector<LogInfo> login` :一个储存了登录栈中用户信息的 `vector`.

成员函数:

`SwitchUser`: 将新的用户信息插入到 `login` 的末端, 新插入的用户选定图书为空。

`Select`: 设置 `login` 末端用户的选定图书。

`logout`: 删除 `login` 末端用户。

```

class LogStatus{
public:
    LogStatus()=default;
    ~LogStatus()=default;
    void SwitchUser(const char *UserId,AccountFile& account_file,TransactionLog &,const char *password="");
    void logout(TransactionLog &,AccountFile&);//撤销最后一次成功执行的 su 指令效果
    void Select(const char *ISBN,BookFile& book_file); //让栈尾的用户选择图书
    std::vector<LogInfo> login;
};

```

log.h

TransactionLog 类:

成员函数:

- `ShowAll`: 展现所有交易的总收入与总支出。
- `Show`: 展现最近 `count` 笔交易的总收入与总支出。
(展现格式: `+ [收入] - [支出]`)
- `log`: 打印营业日志。

```

class TransactionLog{
public:
    TransactionLog();
    ~TransactionLog();
    void cost(long long); //negative
    void earn(long long); //positive
};

```

```
void ShowAll() const;
void Show(int count);
void AddRecord(const char* command ,const char* UserId,int pri);
void AddRecord(const char* command);
void log(LogStatus&);
private:
    int transaction_num;
    int record_num;
    std::fstream iof;
    std::fstream iof2;
    long long total;
    long long earn_total;    //positive
    long long cost_total;   //positive
};
```

文件存储说明

`BookFile` 与 `AccountFile` 中图书与用户的信息均使用块状链表存在二进制文件中, `Logstatus` 中登录信息以 `vector` 存在内存中。

块状链表 `UnrollLink` 类的信息存储:

`file` 文件: 按照顺序存储各个关键词与值的结构体 `data`.

`other_file` 文件: 在头部存储总块数, 然后以此存储各个块的信息结构体 `block_node` .