

# Exchange Simulator

## 设计文档

516030910101

罗宇辰

## 一、实现的功能

### 1. 连接一个服务器与一个客户端

### 2. 客户端可以

- ( 1 ) 发送一个新的 Fix Message 给客户端
- ( 2 ) 发送取消最近一个 order 的消息给客户端
- ( 3 ) 收到、显示并存储客户端发回的消息
- ( 4 ) 检查某一客户所有订单的记录信息

### 3. 服务器可以

- ( 1 ) 接收新的 Fix Message 并回复确认信息
- ( 2 ) 对一个 order 进行 partial fill 或 full fill , 并回复相关记录信息
- ( 3 ) 根据客户端的要求取消某一订单或者返回取消订单失败的信息

## 二、文档介绍

### 1.代码文件

-client.cpp

客户端启动文件

-order.h & order.cpp

客户端函数文件，实现对客户信息的提取和整理，以及对交易

记录的存储、读取和翻译

-server.cpp

服务器端启动文件

-message.h & message.cpp

服务器端命令类和函数文件，实现对收到的信息进行分析 and 记录，并进行 fill 和 cancel 等操作

## 2.TXT 文件（运行中生成）

（1）客户端

-xxxxx.txt （xxxxx 为 CLIENT ID）

记录了对应客户的 order 基本信息，用以提取最新 order 进行 cancel 操作

-xxxxxrecord.txt （xxxxx 为 CLIENT ID）

记录了对应客户的所有交易记录，供用户进行查阅操作

（2）服务器端

-nnnnn.txt （nnnnn 为 GOOD ID）

记录了某种货物的所有能被交易的 order 信息，用以比对进行 fill

-Record.txt

记录了所有交易信息

-Finish.txt

记录了所有已完成的 order 信息

### 三、实现方法介绍

#### 1.客户端

( 1 ) 开启客户端 , 询问要进行的操作【函数 get\_order】

( 2 ) 发送新订单【函数 : new\_order】

-要求用户输入有关信息

-生成 order , 并记录在对应 clientIDrecord.txt 中

( 3 ) 查阅记录【函数 : check\_order】

-要求客户输入其 ID

-读取对应 clientIDrecord.txt 文件内容 , 翻译后显示【函数 : set\_map & rreplace】

( 4 ) 取消订单【函数 : cancel\_order】

-要求客户输入其 ID

-读取 clientID.txt 文件中最新的内容 , 生成 order , 并记录在对应 clientIDrecord.txt 中【函数 : save\_order】

## 2.服务器端

( 1 ) 对收到的信息进行分析，判断请求类型【函数：check】

( 2 ) 处理新订单

-生成 orderID，生成回复信息

-在 order 类中分析记录下 order 包含的信息

-根据 goodID 检索对应 goodID.txt，生成包含可交易 order 的容器【函数：initialize】

-在 order 类中实现 fill

-全过程中将各项交易记录在 Record.txt 中【函数：record】

-向客户端返回交易记录

-将已经完成的 order 记录在 Finish.txt 中，未完成的已更新的 order 重新写入 goodID.txt 中【函数：save】

( 3 ) 取消订单

-在 order 类中检查和实现

## 四、总结和分析

1. 我认为 client 这边没有设计类来封装信息这一点很遗憾，然后因为开始做的比较早，后来学的很多知识都没有用上，改也来不及了，感觉很多地方用的方法都很暴力，一点都不巧妙

2. 因为还是不懂这个 boost 库到底是怎么回事，也不懂多个端口怎么连接，还不懂怎么多次收发消息，所以只做到了 hw2，而且客户端和服务端只能交流一次，非常非常遗憾