

GO 语言聊天室开发文档

张嘉乐 518030910319

2019 年 7 月 26 日

目录

1	聊天室功能说明	1
2	开发帮助	2
2.1	项目基本框架	2
2.2	两个应用程序通信的原理	2
2.3	服务端结构	2
2.3.1	主函数	3
2.3.2	单客户主服务函数 <code>serveIt()</code>	3
2.3.3	客户各种请求函数	3
2.3.4	服务端数据结构	3
2.4	客户端结构	3
2.4.1	Tcp 通讯模块 <code>Client</code> 类	3
2.4.2	主窗口 <code>MainWindows</code>	3
2.5	各个功能类	4
2.6	服务端用户端通信协议	4
3	总结与展望	4
3.1	文件传送	4
3.2	客户信息加密建库	4

1 聊天室功能说明

本项目用 GO 语言搭建服务端，Qt 编写客户端实现不同应用程序之间的通信，以达到聊天功能。其主要功能有：

1. 客户端注册，登录，非注册快速登录
2. 客户端发送消息，群发给连接到的所有用户
3. 客户端通过秘籍成为 VIP
4. VIP 客户可以禁言聊天室的成员。
5. VIP 客户可以解禁言聊天室的成员。

相关重要处理有：

6. 服务端多线程，使用同一连接收发信息时使用互斥锁保证操作原子性。
7. 客户端使用信号槽机制实现收发消息的并行，通过打断信号槽连接保证操作原子性。
7. 客户端界面美化。
8. 客户端界面布局，实现窗口任意缩放。

2 开发帮助

2.1 项目基本框架

聊天室分为服务端和客户端，客户端需要用于用户交互的图形界面模块以及和服务端收发消息的模块，服务端需要有接受用户端连接，登录，发消息的模块以及将消息群发的模块。本项目中服务端用 GO 实现，利用自带的 net 库，方便多线程；客户端用 Qt 实现，使用 Tcpsocket 对象进行连接，读，写。

2.2 两个应用程序通信的原理

通信是通过端口实现的，GO 服务端 net.listen 指定监听的端口并生成监听对象，Qt 客户端 QTcpsocket.connectToHost 指定同一网络中的 IP 和端口后，服务端 accept() 接受请求生成连接对象，服务端就可以通过这一连接对象进行读写，客户端可以通过 QTcpsocket 对象进行读写，两者就建立了通信。

2.3 服务端结构

服务端有三级结构

2.3.1 主函数

主函数建立对端口的监听后循环等待连接，每接受一个连接，就 go 一个新线程函数 `serveIt()` 处理这个连接的所有信息。

2.3.2 单客户主服务函数 `serveIt()`

循环等待客户端发来的请求，并根据请求的类型—登录、注册、发信息、发图片，到具体函数中进行处理。

2.3.3 客户各种请求函数

对于各种请求，服务端和客户端都会进行多次握手确定信息的细节，对于只针对一个客户端的注册，登录请求比较简单，只需确定信息，更新服务端数据库即可。对于群发信息，由于多线程使用同一个连接进行读写，要区别握手信息与新请求，这里使用一个全局字典作为互斥锁来处理。

2.3.4 服务端数据结构

服务端定义了两种结构体和五个字典，结构体用于存放用户信息，分别包括用户名，连接对象，密码中的两项，分别用于验证登录和在线交互。字典有注册的所有用户，在线用户，VIP 用户，被禁言用户，用户连接互斥锁五个。

2.4 客户端结构

客户端有三个模块

2.4.1 Tcp 通讯模块 `Client` 类

使用 `QTcpSocket` 实现对服务端的连接和读写，并提供公共接口给整个应用。

2.4.2 主窗口 MainWindows

定义了所有功能，包括连接服务端，登录，发送信息，充值 VIP，禁言，解禁，并设计相应的 Ui。

2.5 各个功能类

包括连接服务端，登录，注册等等，有些有相应的 Ui 设计。

2.6 服务端用户端通信协议

所有通信均采用多次握手方式，具体应答随功能不同而有差异。

3 总结与展望

3.1 总结

本聊天室实现了不同应用程序之间通信的基本功能并且有用户管理系统，加入了一些功能以增加可用性和趣味性。客户端界面友好美观。

服务端实现了基本功能和附加功能的支持，但效率和稳定性欠佳，在多用户和恶意攻击条件下容易崩溃，需要优化。

3.2 进一步开发

3.2.1 文件传送

服务端和客户端中均已有文件发送的 UI 和作，由于一些问题发送没有成功，后继的开发者可以在此基础上实现

3.2.2 客户信息加密建库

服务端的数据没有读写文件，适用于一次开启后不再关闭的情况，如需维护服务端并且保留用户数据则需要连接后台数据库管理软件。用户密码可以加密后存储