

# 网络聊天室任务说明书

为减轻任务量，降低和区分项目难度，统一网络聊天室的功能细节，方便横向对比和评分，现对本实践中的网络聊天室任务做出以下说明。

## 一、基本功能

本次实践所实现网络聊天室应由一个服务端和多个客户端组成；在用户角色上，所有客户端均为普通用户，没有管理员等角色；在数量上，整个系统运行时仅允许存在一个由服务端维护的聊天室，客户端只能连入和退出聊天室，不能创建聊天室。

系统中服务端和客户端所应具有的具体功能如图 1 所示。

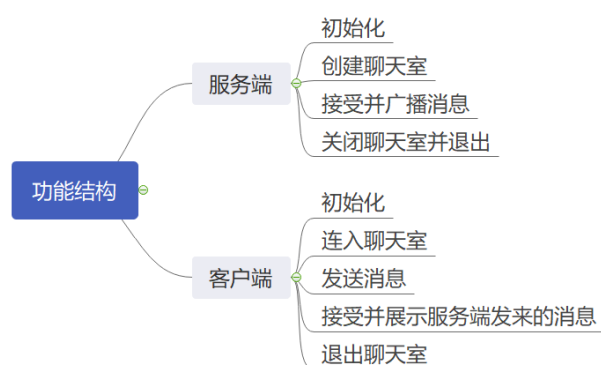


图 1 系统整体的功能结构

为实现以上功能，系统各部分所应维护的基本信息如图 2 所示。

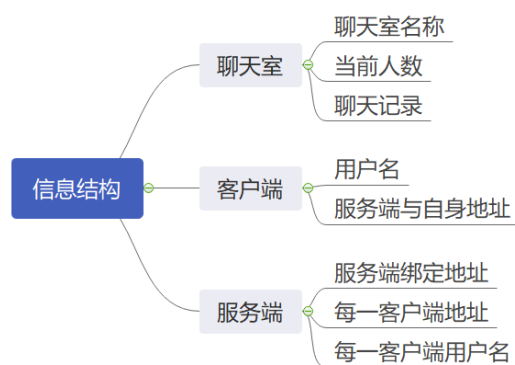


图 2 系统的信息结构

下面，将给出服务端和客户端的工作流程，以展示上述各项功能的运作方式。

图 3 给出了服务端的工作流程。

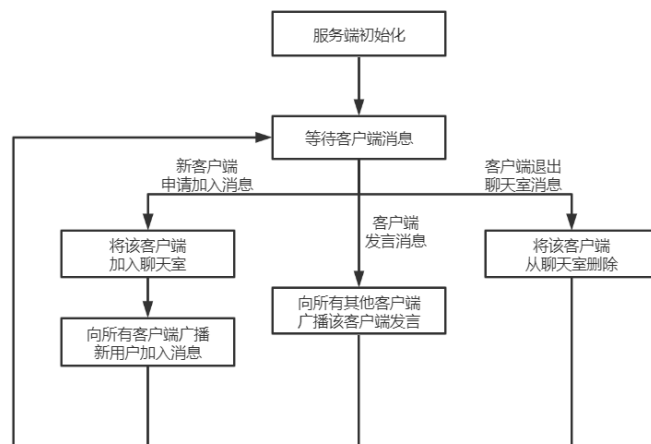


图 3 服务端工作流程

- 服务端初始化时，应根据预设的 IP 地址和端口进行监听，等待客户端连入。
- 服务端应使用 IP 地址、端口或其他自行设置的标识符识别和区分用户，不应使用用户名，因为用户可能重名。
- 服务端只负责聊天消息的接受和转发，不需要保存聊天记录。
- 服务端在任何时间收到关闭聊天室指令后，需要停下当前任务，向所有客户端发送关闭通知，之后结束运行。

本实践鼓励使用不同的设备模拟真实的聊天室情景，但不禁止在一台设备上使用不同的进程和线程模拟不同用户。

图 4 给出了客户端的工作流程。

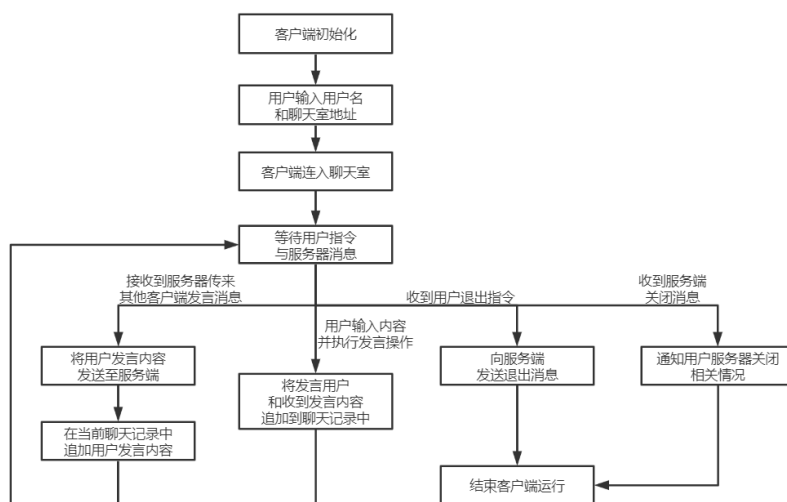


图 4 客户端工作流程

- 客户端初始化时，需要用户输入自己的用户名，以及用户希望连入的聊天室地址，之后客户端根据这些设置发出连入的申请
- 客户端只需要记录和展示自身连入之后的消息，不需要展示历史聊天记录

录。客户端退出后，也不需要保存任何聊天记录。

- 聊天消息仅限于文本消息, 每条聊天记录应至少包含发言用户名以及发言内容, 且应简洁清晰。

最后, 本实践目的不在于设计和实现图形化界面, 因此**图形化界面的有无、好坏不参与评分**。如没有图形界面, 需考虑和设计各类信息的展示方式, 聊天信息、系统提示信息应有足够的区分, 避免混乱。

## 二、扩展功能

### 1. 异常处理

#### 粘包检测和处理

理论上 TCP 已经提供了稳定可靠的数据传输服务, 但是 TCP 默认采用了 Nagle 等优化机制, 将多次间隔较小、数据量小的数据合并成一个大的数据块, 然后进行封包一并发送。这样, 接收端就难于分辨出来了。同时, 如果接收端任务繁重, 也会在接收缓存中出现粘包问题。

只要应用层协议设计合理, 接收端就能从各种粘包情况下正确进行拆分和解包。此时, 需要设计情景, 通过运行截图展示粘包的出现, 分析协议设计的方案, 展示接收端是如何根据协议正确处理粘包问题的。

此外, 还可以通过关闭 TCP 相关传输优化算法, 手动清空缓冲区等手段减少粘包情况的出现。

#### 异常断线检测和处理

使用心跳包等手段, 及时发现和处理客户端甚至服务端的异常掉线情况, 及时通知用户相关情况, 保证系统的正常相应。应设计实验, 模拟异常退出、网络掉线等情景, 通过截图给出系统的应对。

### 2、并发调优与压力测试

使用各种技术手段 (比如非阻塞 socket), 降低服务端的进程、线程开销, 从而更高效利用服务端 CPU、内存等资源。需要设计实验, 模拟至少 1000 个用户以上的并发连接和并行聊天过程, 统计内存占用、CPU 使用率、服务端进程、线程数量、CPU 进行上下文切换时间消耗占比、服务端响应时间等指标, 对比优化前和优化后各项指标的变化, 分析和总结所使用并行技术的效果。

### 3、文件传输

客户端不仅可以发送文本消息, 还可以发送文件。该文件会首先上传至服务

端。如果使用了图形界面，其他客户端可以通过点击文件消息来下载，如果没有使用图形界面，其他客户端可以通过输入下载指令进行下载。注意文件下载的过程不应阻碍正常的聊天功能运作。应模拟文件传输场景，通过截图展示该过程的每一步骤。

#### **4、语音聊天室**

类似多人电话或者语音会议，实现多客户端的语音聊天。