

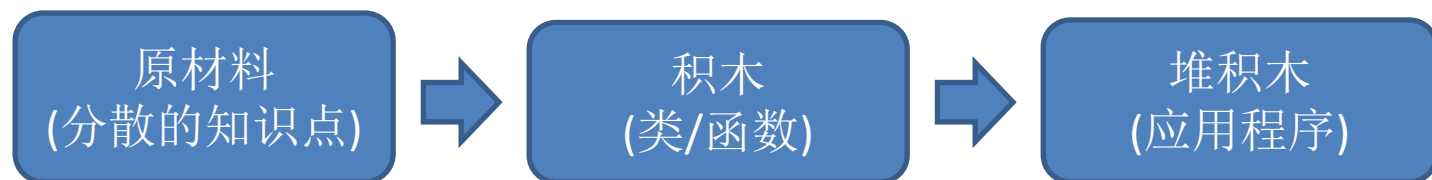
# 《C/C++学习指南》 实战篇

## 006 聊天室

作者 邵发    官网 <http://afanihao.cn>

# 《实战篇》的教学目标

写应用程序就如同堆积木，本篇的重点是展示堆积木的思路... (代码规模并不重要)



- 原材料 语法，线程，SOCKET，SQL，加密解密，标准C库，STL库。。。
- 积木 模块：类/函数
- 堆积木 应用程序：将各模块联合起来

# 本节课的背景知识

《语法篇》： STL

《Qt篇》： QTreeWidget, 定时器,

《补充篇》： 字节编码, AfByteBuffer

《应用篇》： 线程, 互斥锁, socket

# 需求 (要完成的目标功能)

聊天室：

分为服务器端和客户端。

用客户端软件登录后，多个用户可以发言。

服务器端：

- (1) 在线用户列表
- (2) 掉线检测

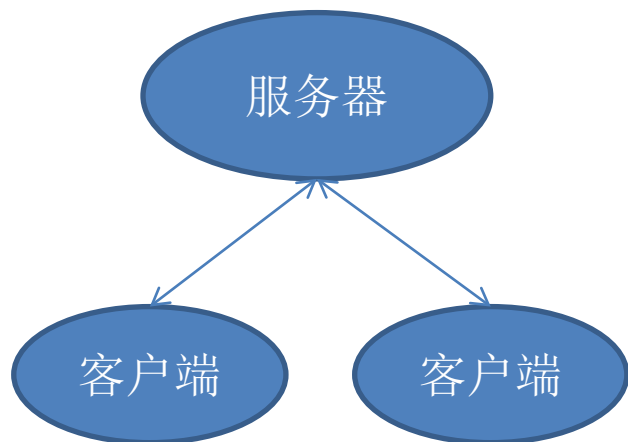
客户端：

- (1) 多用户同时发言
- (2) 消息显示

# 系统 架构

运行一个服务器

同时运行多个客户端



# 服务器端

RecvTask

RecvTask: 接收客户端的消息。

SendTask

SendTask: 处理消息列表。将消息分发之所有客户端。

主界面

主界面: 显示当前登录的用户

# 客户端

SendTask

RecvTask

主界面

**SendTask:** 负责将消息发送给服务器。

**RecvTask:** 负责接收来自服务器的消息。

主界面:

发送: 将用户输入打包, 交给  
**SendTask**

接收: 从**RecvTask**取得消息, 显示  
到界面。

# 技术点(1) 消息封装

借助AfByteBuffer完成

字段	类型	意义
MsgType	UINT16	消息类型。 MSG_JOIN: 心跳消息 MSG_TEXT: 文本消息。
UserName	STRING	用户名
Text	STRING	消息文本。



# 技术点(2) 心跳检测

服务器如何检测客户是否掉线？

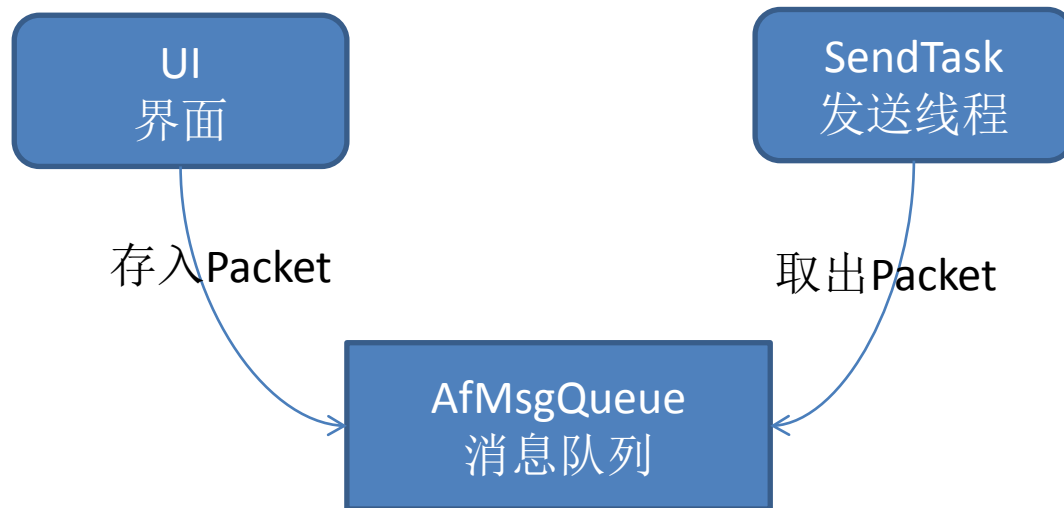
心跳机制：(HeartBeat)

约定客户端每隔1秒钟发送一个心跳HB到服务器。。。服务器记录并刷新每个客户端的最后一个心跳的时间... lastActive...

服务器定期检测lastActive，或超过10秒则认为此客户端已经掉线。

## 技术点(3) 消息队列

比如，对于客户端来说，界面如何将消息包交给SendTask发送？（互斥机制）



# 小结

相关技术点较难，在实际项目中可以参考。