Sockapi设计方案

1. 整体设计
   1. 简介

本文主要描述一套socket API接口，通过poll模型实现socket的快速读写及处理操作，用户仅需添加/删除服务接口，即可实现底层socket消息自收发。

本项目软件对外提供C/C++动态库和接口头文件，封装了底层通信框架，由用户实现应用层业务逻辑。

1. 流程原理
   1. 框架流程

本项目软件的整体框架主要包含三个处理线程：

数据接收线程：底层socket的接收探测和消息读取线程，包含超时和流控机制。

数据发送线程：底层socket的消息发送线程，包含超时和流控机制。

消息处理线程：所有socket消息及控制消息的处理线程，按照FIFO的串行处理机制。

另外，每个socket都有各自的接收队列，发送队列，消息处理队列，严格保证各自socket的消息时序；但是不同socket之间不保证消息的严格时序。

1. 数据接口
   1. 数据设计

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 类名 | 功能描述 |  |
| 1 | SockFrame | 框架类，唯一对外接口类 |  |
| 2 | SockProto | 协议类，由外部提供消息解析 |  |
| 3 | ISockSvr | 服务端类，由外部注册 |  |
| 4 | ISockCli | 客户端类，由外部注册 |  |
| 5 | NodeMsg | 消息头类，内部隐藏细节，提供外部调用 |  |
| 6 | SockBuffer | 读消息结构体，用于读缓存。 |  |
| 7 | MsgHead\_t | 示例消息头 |  |

* 1. SockFrame功能说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 接口描述 | 接口标识 |  |
| 1 | 添加服务端 | creatSvr |  |
| 2 | 添加客户端 | creatCli |  |
| 3 | 延迟添加客户端 | sheduleCli |  |
| 4 | 定时任务 | schedule |  |

1. 约束规则