

黄子轩

15918551668 | 1750380174@qq.com
C++开发 | [kigsont \(github.com\)](https://github.com/kigsont)



教育经历

广东金融学院

2020年09月 - 2024年06月

软件工程 本科

相关课程：C++程序设计，设计模式，计算机网络

CET-4

专业技能

C++

熟练 C++11/14/17 标准，了解 C++20，并能灵活应用于实际项目中
熟练使用智能指针、Lambda 以及 STL 常用容器，能够高效地进行数据处理和算法实现
具备熟悉的 C++ 并发编程能力，熟练使用多线程编程组件，能够设计高性能、高可靠性的并发程序
熟悉常用的 C++ 库，拥有丰富的项目实战经验

其他

熟悉常用 数据结构 和 算法
熟悉 操作系统原理(并发、虚拟化、持久化)
熟悉网络编程，包括 select、epoll 等 IO复用模型，熟悉 TCP/IP 通信原理，熟悉 TCP、HTTP 等常用协议
熟悉 MySQL 数据库(如索引，事务)，熟悉 Redis 的基本数据结构及常用功能
熟悉 Golang 编程语言，了解 GC、GMP 原理，熟悉 Python，了解 Java 编程语言
熟练掌握 Qt 框架和 Widgets 组件，了解 Qt 的各种组件及其信号槽机制，能够快速开发跨平台应用程序
熟练使用 Git 版本控制工具，熟练掌握 gdb、gcc、cmake 等 Linux C++ 开发工具链，具备良好的代码调试能力

实习经历

大族激光科技产业集团股份有限公司

2023年06月 - 2023年08月

C++开发实习生

负责公司商业项目的后端开发、设备通信协议数据交互程序开发、项目维护与优化工作

项目经历

Linux下的网络库 - TinyWebLib

2022.11

项目简介： TinyWebLib 是一个独立开发重写muduo库的高并发和实时性的网络库。在C++开发中muduo库依赖较多，为提升日常开发效率且为保证数据安全传输，重新编写基于 C++17 的 muduo库，去除了muduo库的boost依赖，并添加了TLS安全通信功能。该网络库实现了事件驱动、定时器、线程池和独立的异步日志模块，保留了muduo库的核心特性，同时融入了最新的C++ 标准和安全通信功能，为开发者提供高效、安全的网络通信解决方案。

主要技术：

- 使用 Linux 下 epoll 的 Level Triggered 模式配合非阻塞 socket，根据 Reactor 模型编写项目主体
- 根据 one loop per thread 线程模型理念，实现管理 EventLoop 的线程池
- 使用 fmt 库重新编写异步日志模块，但保留 muduo库异步日志的双缓冲机制，前端写入，后端落盘
- 使用 Linux 下 timerfd 实现定时器功能，并使用基于红黑树的 C++标准库 set 容器管理定时器队列
- 使用 OpenSSL库的 TLS1.3 非对称加密安全通信代替普通的 socket 传输数据
- 使用状态模式解析HTTP报文，能根据该库简易编写高效的 http server
- 使用 Benchmark 对 web server 进行压力测试，并用 Perf 生成火焰图进行性能分析

可跨语言调用的TinyRPC框架

2023.1

项目简介： 为了实现简单易用的RPC框架，独立完成基于 tinyWebLib + protobuf + zookeeper 的非阻塞 rpc 框架 tinyRPC。旨在实现易用、高可用的数据传输和处理。用该 RPC 框架实现的 C++RPC 服务器支持跨语言调用，而用该框架实现的C++客户端支持阻塞/非阻塞 RPC调用，同时内嵌线程池方便多线程开发。

主要技术：

- 使用 protobuf 库设计 Remote Procedure Call 的传输报文格式
- 使用 C++标准库的 future<void> 实现事件通信，实现 rpc 非阻塞调用
- 编写基于concurrentqueue库无锁队列的支持临时线程扩展的线程池，以支持异步 rpc 调用
- 对每次 rpc 调用的报文大小进行分析，解决 TCP “粘包” 导致数据包信息不全问题
- 使用 zookeeper 实现 rpc 服务的注册与获取
- 使用 C++17 标准库的容器 variant 编写 json解析器去解析配置文件
- 使用 Golang 编写 客户端测试程序调用 C++服务端的 rpc服务，完成 rpc 跨语言调用测试

基于Qt的网络对战游戏 – BattleShip

2023.3

项目简介： BattleShip 是我第一个独立开发的游戏、一款基于Qt框架开发的战争策略会和对战游戏，支持单机和网络对战模式。玩家可以在棋盘上布置自己的战舰，并与其他玩家进行对战。该游戏还提供了强大的社交功能，包括私聊、群聊和邀请对战。服务器程序由TinyWeb Lib 网络库开发而成，负责基于json报文的大规模实时游戏玩家间数据的解析和处理。

主要技术：

- 使用 Qt基础组件，实现Client端的登录、注册、好友展示和游戏界面
- 设计数据格式以表示战舰在棋盘上的分布，实现游戏的展示
- 设计游戏对战场景，编写相关组件，重写对应需要处理的事件，实现游戏逻辑
- 编写服务器业务逻辑代码，设置对不同报文的处理方式
- 设计玩家相关信息的mysql数据库表，进行数据的保存
- 在服务端使用mysqlclient库连接mysql数据库，并使用连接池提高性能
- 基于 OpenSSL，客户端使用DES对称加密持久化玩家密码，服务端保存SHA256加密的玩家密码
- 使用 基于发布-订阅的 redis 消息队列，实现跨服务器的消息通信
- 配置 nginx 的 tcp负载均衡，实现服务器的集群功能

校园经历

- 计算机爪哇部落后端 C++组组长
- 为学校爪哇部落提供 tinyWebLib 和 tinyRPC 项目
- 参加学校组织的软件开发大赛，使用 C++编写 IM 聊天室