

求职意向 - C++后台开发工程师

● 基本信息

👤 徐泽坤

🎓 桂林电子科技大学(本科)

📖 软件工程专业(2027届)

🏠 广西桂林

📞 14748487395 (微信同)

✉ 3293485673@qq.com

🌐 <https://blog.csdn.net/TTKunn>

🔗 <https://github.com/TT-Kun/>

● 专业技能

编程基础/数据结构/设计模式/数据库

- 熟悉C/C++基本语法，了解面向对象思想及封装、继承、多态、模板编程等。
- 熟悉C/C++动态内存管理操作，了解智能指针及RAII思想。
- 熟悉常用 STL 容器的使用并了解常用容器和空间配置器的底层原理。
- 熟悉基本的数据结构，如：链表、栈、队列、堆、树（二叉树,AVL数,红黑树等）、图、哈希表等
- 掌握字符串、栈和队列、前缀、排序、贪心、回溯、动态规划等基础算法。
- 熟悉常见设计模式，如：单例（懒汉,饿汉）、工厂（简单工厂,工厂方法,抽象工厂）、建造者模式等。
- 熟悉 MySQL 数据库的CRUD操作，掌握复杂查询及索引优化策略，熟悉视图机制与设计范式等。
- 熟悉使用Qt C++进行简单的桌面端应用开发。
- 了解Docker、Nginx等中间件，并能够使用docker进行项目的容器化部署、Nginx部署前端网页。

操作系统 / Linux 系统编程

- 熟悉常用 Linux 命令及 vim、gcc、gdb、git、shell等工具的使用及 Makefile 的编写。
- 熟悉 Linux 下多线程、多进程编程，了解进程的同步与互斥和信号的处理机制。
- 熟悉 Linux 下进程间通信机制，如：管道、消息队列、信号量、共享内存等。

计算机网络 / Linux 网络编程

- 熟悉常见的网络通信协议，如：TCP、UDP、HTTP 等。
- 熟悉网络编程基础，熟悉 Linux 下 Socket 编程。
- 熟悉TCP三次握手，以及TCP粘包问题的解决方案
- 了解 TCP/IP 协议栈和数据的封装与分用。
- 了解常见 I/O 模型，如：阻塞、非阻塞、信号驱动、多路复用、异步。
- 了解多路复用 I/O 技术，如：select、poll、epoll。

● 项目经历

基于C++ / Linux 实现的分布式RPC服务注册与调用系统

所用技术：

linux C++异步操作 并发编程 json序列化 muduo网络库 RPC

项目介绍：

通过学习gRPC、Thrift等开源RPC框架以及网络通信高性能服务器等相关知识，实现了基础的RPC远程调用功能，以及基于服务注册和发现的RPC远程调用功能

主要工作：

1. 自定义通信协议：解决TCP粘包/拆包问题，通过自定义消息格式、编解码器实现数据完整传输；使用JsonCpp实现Json格式数据的序列化与反序列化，保证数据传输的规范性和正确性。
2. 高并发网络模型：基于MuduoReactor网络模型实现网络层，实现网络IO与RPC方法调用的解耦设计，提升可维护性。
3. 框架设计：项目采用抽象层、实现层、业务层三层架构设计，通过抽象网络通信和协议组件实现低耦合高可维护，降低框架依赖并提高框架灵活度和可维护性。

代码仓库：<https://github.com/TT-Kun/JsonRPC>

基于Qt/C++开发的多媒体播放器开发项目

所用技术：

Qt 6.5.3 C++ QMediaPlayer 事件过滤器 信号-槽机制 自定义窗口框架 QSS样式表

项目介绍：

学习Qt多媒体技术与GUI编程，实现一个能够支持常见媒体格式播放的，包含倍速播放、播放记录、主题切换、进度条预览等多种基础功能的，具有自定义无边框界面的多媒体播放器应用。采用模块化设计思想，将播放器功能分解为多个独立且相互协作的组件，实现了较为全面的媒体播放控制功能。

主要工作：

1. 设计并实现了基于QGridLayout的自定义无边框窗口以满足标题栏的样式设计，支持窗口拖拽、缩放等功能
2. 基于QMediaPlayer等设计实现媒体控制模块，实现播放、暂停、进度控制及预览缩略图等播放核心功能
3. 实现了基于QListWidget等控件的播放列表管理功能，支持多种播放模式（顺序、循环、随机等）
4. 使用Qt信号-槽机制和事件过滤器优化组件间通信，降低了模块间耦合度
5. 采用QSettings框架实现配置持久化，保存播放历史、窗口状态等用户设置
6. 通过Qt样式表(QSS)实现了可切换的明暗主题，提升用户体验

代码仓库：https://github.com/TT-Kun/muti_player

基于C++ / Windows 实现的高并发内存池

所用技术：

C++ 并发编程 内存管理 Windows系统库 哈希表 双向链表 多级缓存 页表映射

项目介绍：

学习并发编程与内存管理相关知识，参考学习谷歌TCMalloc内存池结构设计和相关知识，自主实现了基础的高并发内存分配与回收功能，提高了多线程高并发场景下内存申请与释放的性能。

主要工作：

1. 分层内存架构设计：设计实现了线程级缓存、中央共享缓存和页级堆管理的三层架构，采用多级缓存机制提升性能，减少系统调用次数；利用自定义内存分配算法提高内存利用率和分配速度。
2. 高并发机制实现：基于线程本地存储(TLS)技术避免频繁加锁，减少线程间竞争；对于共享资源使用互斥锁保证线程安全，设计并实现了高效的自由链表管理算法。
3. 核心结构设计：实现基于双向链表与多重哈希桶的内存块管理系统；设计多级映射结构结合压缩前缀树(Radix Tree)实现O(m)复杂度的地址查找；通过精细化大小类分类与动态分割合并算法优化内存碎片，构建平衡分配效率与空间利用率的自适应内存管理系统。

代码仓库：<https://github.com/TT-Kun/high-concurrent-memory-pool>

其他项目

1. 基于Qt6、vue3、python开发的碳足迹计算器，包含Web网页端以及Qt桌面端
2. 基于Qt6开发的建议校园导航应用
3. 基于通信UOS操作系统和Qt5开发的智能多媒体播放器
4. 基于vue3、springboot、mysql开发的学生宿舍管理系统