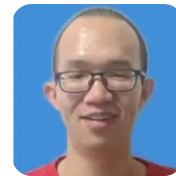


# 白金陇

男 | 24岁 | 15754512820 | 2958187064@qq.com  
C++



## 教育经历

哈尔滨工程大学 硕士 计算机科学与技术 2023-2026  
哈尔滨工程大学 本科 机械设计制造及其自动化 2019-2023

## 专业技能

- 掌握 C++ 编程语言，具备面向对象编程（OOP）基础。熟练运用 C++ STL 容器、算法和迭代器，提升代码效率与可读性。掌握 C++ 多线程编程，运用线程、互斥量和条件变量实现并发编程，有良好线程安全意识。
- 掌握 epoll、poll、select 网络编程模型。熟悉 Reactor 模型，可应用于实际开发。掌握 TCP/IP 协议，了解 nginx 服务器与 muduo 网络库，能进行高效网络编程。
- 熟悉 MySQL 数据库设计、优化与 SQL 编写，掌握 Redis 缓存数据库基本数据结构与使用场景。
- 熟悉 go 语言，能使用 go mod 进行环境配置和项目搭建，能进行简单的模块开发。
- 熟悉深度强化学习基本概念与 MADDPG、PPO、DDPG、A3C 等算法，能用 PyTorch 实现简单强化学习模型。
- 了解 Docker 容器技术，可创建、管理和部署容器化应用，掌握 Dockerfile 编写与镜像管理。
- 熟练使用 GDB 调试工具，利用断点、观察点和堆栈跟踪进行程序调试与性能分析。
- 掌握 CMake 构建系统，编写和维护 CMakeLists.txt 文件，熟悉跨平台构建与依赖管理。
- 熟悉 Linux 常用命令行进行系统管理、文件操作和网络配置，具备编写 Shell 脚本的能力。
- 熟悉使用 Git 进行版本控制，可进行代码分支管理、合并及冲突解决。
- 了解 FFmpeg 多媒体处理框架，可进行音视频编解码、格式转换和流处理。

## 项目经历

多水下机器人协同编队避障 负责人 2024.06-至今

### 内容:

项目基于视景仿真系统,旨在通过c++实现多个水下机器人的编队组成与维持算法,特别是在向目标点行驶过程中进行编队协同避障。项目涉及的主要障碍物包括潜艇和潜标。

### 主要工作:

- 设计并实现了多机器人编队算法，结合协同避障算法，确保机器人在复杂环境中有效避开潜艇、潜标等障碍物，提升编队的稳定性和灵活性。同时，通过并行处理规划、感知和控制模块，采用多线程优化系统响应速度和处理能力，使机器人能够快速适应动态环境变化。
- 基于UDP和TCP协议实现模块间数据的高效传输，并利用RapidJSON库实现通信数据的序列化和反序列化，简化数据收发流程，支持实时通信需求。
- 使用Docker容器化搭建跨平台开发环境，简化开发和部署流程。结合CMake统一管理代码编译和依赖关系，并确保项目跨平台可移植性。
- 通过Git进行代码版本管理，结合分支管理和合并请求流程，提升团队协作效率。

个人收获：深入理解并掌握了多线程编程和网络编程的核心概念与实践。熟悉了项目的版本管理流程，提升了团队协作能力。提高了使用GDB进行代码调试的技能,能够更有效地定位和修复代码中的问题。

### 业绩:

在沈阳自动化所的仿真测试项目中，负责协同编队避障算法的优化与测试，成功实现了98%的避障率，超出设定的90%目标，显著提升了系统的安全性和可靠性。

基于多智能体深度强化学习的自主水下机器人分组 负责人 2024.01-2024.06

本项目旨在基于大规模自主水下航行器（AUV）集群的初步任务分配，在Unity仿真环境下研究如何结合有限的感知能力，实现同一子群内多台 AUV 的自动编队和分组。

## 主要工作：

- 构建了一个**多智能体深度强化学习框架**，以支持**不同角色**的AUV在复杂环境中的协作。
- 开发了适应**弱通信条件**的**环境感知模型**，以提高AUV在信息受限情况下的决策能力。
- 提出了一种**多阶段目标点选择策略**，以优化AUV的航行路径和任务执行效率。
- 在设计中，动作空间仅设置为**艏向加速度、水平转动速度和竖直方向转动速度**，以符合水下机器人的物理特性。
- 不同角色的AUV设计了相应的**奖励函数**，以促进有效的协作和任务完成。
- 利用**Unity**和**ML-Agents库**，成功实现了上述算法，并进行了仿真测试。

个人收获：我深入学习了深度强化学习的相关理论与应用，掌握了**Unity**的基本功能和**ML-Agents库**的使用。这些经验不仅提升了我的编程能力，还增强了我对多智能体系统和水下机器人领域的理解。

## 基于c++11重构muduo网络库 负责人

2024.12-至今

本项目在学习和参考**Muduo库**的基础上，使用C++11对其进行了重构，实现了一个基于**多Reactor多线程模型**的高性能网络库。重点实现了**Acceptor、TcpConnection、Channel、Poller、EventLoop、异步日志模块**。

### 主要工作：

- 采用**non-blocking（非阻塞）+ io-multiplexing（I/O多路复用）+ one loop per thread**的多线程服务器网络编程模型，结合**Reactor模式**实现高效事件处理。
- 遵循**RAII原则**，使用**智能指针**对**Poller和Channel**等资源进行自动管理，使用**atomic（原子操作）**保护状态变量，避免竞争条件。去除了对Boost库的依赖，完全基于**C++11标准库**实现。
- 参考**Netty**的设计思想，将缓冲区划分为**prepend、read和write**三个标志区域，提升数据操作效率和逻辑清晰度。
- 基于**双缓冲区实现异步日志**，由后端线程向磁盘写入前端线程的日志信息，避免数据写入磁盘是阻塞网络的问题。

个人收获：对经典的**五种io模型**和**Reactor模型**有了深刻认识，强化了**面向对象编程**的设计理念和实践能力，熟悉了C++11标准库的使用方式。

## 多线程高并发通信框架 负责人

2024.09-2024.12

本项目旨在使用C/C++从零构建一个**多线程高并发服务器程序框架**。该框架设计为可扩展的，用户只需添加业务逻辑，即可用于开发多种服务器项目，如通信服务器或网络游戏等。

### 主要工作：

- 模仿**Nginx的事件模块**，使用**epoll**实现高效的事件驱动编程模型，优化服务器的I/O性能
- 按照**包头+包体**格式正确接收客户端发送的数据包，解决数据粘包问题，确保数据的完整性和准确性，根据接收到的数据包类型，执行相应的业务处理逻辑，并自定义封装应用层逻辑协议，以满足不同业务需求。
- 采用**线程池模式**处理业务逻辑，提高服务器的并发处理能力，并将业务处理产生的结果数据包正确返回给客户端，
- 实现**连接池管理**，采用**延迟回收技术**，消除导致服务器不稳定的因素，提升系统的整体性能和可靠性。

个人收获：深入理解了**epoll**的使用流程和工作原理，掌握了高效的**事件驱动编程模型**。熟悉了**信号处理、日志打印、子进程管理和守护进程**的实现，增强了对操作系统底层机制的理解。

## 个人优势

拥有**团队合作**项目的协作开发经验，善于合理安排时间并协调各方任务，积极参与团队讨论，促进了团队成员之间的有效沟通和合作。

## 资格证书

CET-6 CET-4

## 荣誉奖项

校级一等奖学金、校级二等奖学金