

王志鹏

github.com/Zip000000

(+86) 15141783640

p307110017p@gmail.com 307110017@qq.com



教育背景

2016 年 9 月 - 2020 年 6 月 本科, 软件工程专业, 哈尔滨工程大学, 哈尔滨

相关技能

编程技能: 熟悉 C/C++, Shell 脚本语言, 了解 Python

平台: 熟悉 Linux 环境

数据结构与算法: 熟练掌握 链表, 栈, 队列, 树, 堆等基础数据结构; 以及单调栈, 单调队列, 优先队列, 哈弗曼编码, 二叉树的线索化; 并查集 (包括: Quick-Find 算法, Quick-Union 算法, Weighted Quick-Union 算法); 字符串匹配算法 (包括: KMP 算法, Sunday 算法, Shift-And 算法, 字典树, 双数组字典树, AC 自动机); 查找算法 (包括: 二分查找, 三分查找); 树状数组与线段树;

项目经历

2019 年 6 月 Linux 部分系统命令的 C 语言实现, 个人项目, C/Linux 编程

- > ls: 实现了根据窗口大小与文件名长度的智能分栏, 并按文件名排序, 输出结果整洁美观。
- > ls -al: 实现了与系统命令几乎相同的效果, 能够输出文件权限, 用户名, 组名, 文件大小, 修改时间, 文件名等信息, 并按序输出。
- > cp -a: 实现对任意类型文件的复制, 支持一次复制多个文件到指定文件夹, 并支持直接复制文件夹。

C Linux

2019 年 7 月 基于 Socket 的多人足球小游戏, 个人项目, C/Linux 编程

- > 该项目实现了多人足球对战, 界面采用 curses 库实现, 实现玩家与足球的运动, 实时更新游戏画面, 并能够对每名玩家进球得分进行统计展示。
- > 服务端采用 epoll 处理多人场景, 能够处理多名玩家同时运动与踢球的动作请求, 基于事件驱动型编程。
- > 服务端利用计时器 itimer 与信号实现了足球的减速运动。
- > 底层使用 TCP/IP 协议进行数据传输。

C Linux Socket TCP 多线程

2019 年 8 月 基于 Socket 的 Linux 服务器集群监控系统, 个人项目, C/Linux 编程

- > 该系统可以实现对 Linux 服务器中各项性能指标的实时监控, 并支持数万台 Client 端并发。
- > 基于 Socket 网络编程编写, 并应用 I/O 多路复用技术 (epoll, select), 非阻塞 IO, 采用事件驱动型编程。
- > 服务端采用多线程, 客户端采用多进程, 使用互斥锁保护临界资源, 使用进程间通信 IPC (共享内存等)。
- > 服务端维护存储用户连接的资源池, 并控制相对负载均衡, 并发数可通过配置文件按需调整。
- > 使用心跳机制, 通过应用层心跳来判断双方是否存活, 服务端对掉线用户进行清除, 客户端掉线时主动进行断线重连。
- > 通过系统调用的方式执行 shell 脚本, 从相关系统命令与系统文件来获取服务器状态信息, 主要包括: CPU 信息、用户信息、内存信息、磁盘信息、恶意进程监测, 系统运行概况等。
- > 客户端产生的日志信息首先缓存在本地, 在发送到服务端数据库保存后清除本地数据, 释放空间。生成日志信息时模仿内核回写机制, 减少读写操作消耗资源。
- > 底层使用 TCP/IP 协议进行数据传输, 由服务端主动请求数据。对于长时间积压数据的收发能够有效处理, 解决了粘包问题。
- > 实现将该项目以守护进程的方式启动, 位于后台运行。

C Shell Linux 多进程 多线程 Socket TCP I/O 多路复用

2019 年 8 月 | 面向大规模文本的任意类型数据检索, 个人项目, C/Linux 编程

- › 采用二叉字典树, 支持存储任意类型数据, 优化普通字典树占用内存过多的现象。
- › 利用哈弗曼编码, 针对不同的字符频率, 优化当前场景下的字符编码, 提高效率。
- › 采用双数组字典树保存字典树, 实现字典树的可迁移性, 同时节省内存开销。
- › 采用 AC 自动机实现多模匹配, 实现针对大规模文本的高效检索。

C Linux 哈弗曼编码 字典树 双数组字典树 AC 自动机

2019 年 6 月 | 轻量级自动测试框架, 个人项目, C/Linux 编程

- › 借鉴谷歌开源测试框架, 实现单元测试功能, 并且能一次进行多个单元的测试。
- › 输出界面友好, 显示错误样例, 并统计错误率。
- › 利用大量宏来进行实现, 提高了测试框架的移植性和拓展性, 使得可以在多种场景下进行测试。
- › 开发过程中使用 makefile, 减少不必要的编译。

C Linux makefile

💡 其他

英语: 无障碍阅读英文文档, CET-6

编码统计: 通过 Shell 脚本统计, 4 个月内大约有 **3.8 万行**的代码量