

姓 名: 李 永 辉
性 别: 男

学 历: 本 科
专 业: 计算机科学与技术



18700575796

yonghuilee.cn@gmail.com

Github: <https://github.com/NorthWardTop>

求职意向: C/C++ Linux 研发工程师

教育背景

西安石油大学
计算机科学与技术

2016 - 2020
本科

获得技能

- ☑ 熟练掌握 C/C++, 了解 Python
- ☑ 了解 linux 系统, 注释 0.11 内核
- ☑ 熟悉 linux 多线程进程开发/了解字符设备驱动模型
- ☑ 熟悉 Qt 开发, 熟悉网络编程
- ☑ 熟悉 ARM Cortex A 系列芯片, 了解 Verilog HDL
- ☑ 熟练 gcc gdb make 等工具

获奖情况

- ◇ 2018 年全国大学生数学建模竞赛省赛 二等奖
- ◇ 第十届蓝桥杯大赛软件类 C/C++ 程序设计省赛 三等奖
- ◇ “互联网+”大赛<物联网编程语言 “hur” >校赛主赛道 三等奖

工作经历

湖南潭州教育网络科技有限公司
C/C++ 助教答疑老师

C/C++ 研究院 2017.2 - 2018.5
负责学员线上答疑, 批改作业, 给主讲老师汇报作业情况

北京神州绿盟信息安全科技股份有限公司
Linux C 研发工程师(实习)

RCM 技术部 2019.5- 至今
负责 docker 微蜜罐研发, 网络嗅探程序, 注入程序和 hook 研发

项目经验

- ◇ 项目名称: Linux 高并发线程池服务器
- ◇ 开发环境: Ubuntu 16.04+vscode+pthread+(make gcc gdb)
- ◇ 项目地址: <https://github.com/NorthWardTop/thread-pool>
- ◇ 描 述: 使用三队列方式, 任务等待队列, 线程等待队列, 任务执行队列
使用三个核心管理线程, 任务管理线程通过 socket 接受连接后生产任务放入任务等待队列,
线程管理线程从线程队列取空闲线程, 从任务队列取任务, 放入任务执行队列, 启动执行
线程监控线程负责线程数的动态变化, 客户端连接后可下载文件
- ◇ 获得经验: 熟练掌握线程并发控制, 条件阻塞和互斥锁的配合使用, 学会了模块化和面向对象的设计方法



项目经验2

- 项目名称: **基于 S5P 6818 KEY & LED & ADC 驱动程序设计**
- 开发环境: Ubuntu16.04+vs code+lpthread+(gcc+make+gdb)
- 项目地址: <https://github.com/NorthWardTop/linux-driver-template>
- 描述: 对 LED&KEY 驱动, 应用层使用 ioctl 发送命令到内核层, 内核调用 unlocked_ioctl 函数指针, 对 ADC 驱动, 通过内存映射寄存器地址的方式操作寄存器, 通过 copy_to_user 向应用提供请求的寄存器数据
- 获得经验: 更加熟悉字符设备驱动模型, 熟练从寄存器->内核空间->应用空间数据通路, 熟练寄存器物理地址与内存虚拟地址的转化



个人经历

- 学校职责: 在学校担任易班技术部部长, 带着部员做信息收集系统, 任务已交付, 主要负责项目进度管理, 人员分配, 资源分配和运维
- 关注领域: 熟悉 ARM 架构如 Cortex A9 A53 等处理器, 了解 mips 架构, 熟悉厂商芯片如 S5P6818, S3C2440, Exynos4412 等, 开源代码研究过 uboot 启动过程/Linux-0.11 内核, 了解 MT7620 处理器和 OpenWrt 路由器系统
- 编程方面: 学习 C/C++ 自己实现 vector. 学习 GNU 编程, 熟悉网络编程, 网络数据包封装解包. 熟悉内核模块编程, 字符设备驱动编程. 熟悉多进程多线程编程, 进程通信, 并发控制, 信号量互斥锁条件锁读写锁自旋锁等. 了解 elf 文件结构, 了解进程注入和 hook
- 个人成就: 在自己实现 Linux 高并发线程池, 在 FPGA 上实现 LCD 驱动程序, 在 GEC6818 开发板上实现 ADC 电压转换驱动. 目前在绿盟科技实习, 负责微蜜罐镜像研发(已发布), LAS(Log Audit System)新功能研发如网络嗅探, 进程注入与系统调用拦截



性格爱好

- 我喜欢玩一些智能设备, 如智能路由器, 智能音箱, 我是一个技术控. 自带 8M 的 flash 太小, 自己找 16M 的焊接上去, 我的路由器是 MT7620A 的 CPU, 刷了自己编译的 OpenWrt 固件, 我喜欢我的东西自动化和智能化, 比如将校园网的认证客户端交叉编译并移植到路由器上设置为开机自启, 这样在校园的任何地方, 只要我插上网线, 就能自动认证上网
- 我喜欢健身, 健身能让自己大汗淋漓, 这样很痛快, 每天写代码的疲惫都能放下, 让自己的肌肉非常膨胀就像每天的生活如此充实一样, 偶尔也和朋友骑自行车野行
- 我有很强的代码洁癖, 对变量命名, 模块化, 注释规则, 文档以及软件工程之类的规范特别尊重, 我很注重编码规则和代码质量, linux 下我会修改代码-Wall 到 0 warning, 也很注重代码的性能, 就算我写代码慢, 但是一定写出的都是高质量的性能最强的代码, 我认为写代码最难的就是标识符命名 #_#