

长风破浪会有时，直挂云帆济沧海

- C/C++ 软件工程师的简历制作与自我包装

简历结构

- 个人信息：如实写便于对方公司联系自己
- 教育背景：硬指标，学历高学校强占优势
- 工作履历：实话实说，简明扼要，拉清单
- 专业技能：开发岗的重中之重，越细越好
- 项目经验：目标明确有针对性，投其所好
- 自我评价：补短板，显优势，亮点要突出
- 荣誉证书：有更好，没有亦无妨，加分项

个人信息

- 四个必要信息
- 其它次要信息
- 不需要的信息

四个必要信息

- 姓名：与身份证一致
- 学历：专科、本科、硕士(含在职)、博士
- 电话：约面试
- 邮箱：发Offer

其它次要信息

- 婚况：女性可以写，男性不必写
 - 女性孕产期对求职有一定影响，尤其是已婚未孕
 - 婚姻状况对男性开发岗的影响不大
- 博客：如果有可以加上
- GitHub：对应聘开发岗比较有用

不需要的信息

- 个性、政治面貌、英语四六级、兴趣爱好
- 简洁明了，与找工作无关的信息没有意义

教育背景

- 实事求是，不要造假，也不要含糊其辞，引起误会
- 在职(单证，只有学位证没有毕业证)的本科或硕士，个人信息学历里直接写本科或硕士，这里详细说明

工作履历

- 由近及远据实写
 - xx公司, xx年xx月 - xx年xx月, 职位
 - xx公司, xx年xx月 - xx年xx月, 职位
 - xx公司, xx年xx月 - xx年xx月, 职位
- 职位不是必须的
 - 一般开发岗, 如软件工程师, 不需要写
 - 小(非知名)公司架构师, 主程, 技术总监等, 没必要写
 - 大(行业知名)公司高级管理岗, 可以写
- 工作职责不用写

专业技能

- 三大忌讳
- 一般结构
- 参考范例
- 两点建议

三大忌讳

- 形容词太多：良好、丰富、深入、熟练、精通……文字游戏
- 叙述太空泛：网络通信、数据库、操作系统……没具体内容
- 技能没重点：编程语言、数据结构、算法……凸显不出专业

一般结构

- 编程语言
 - 熟悉C/C++语言，了解STL、Boost，对C++新特性有一定了解，并用于实际工作中，解决实际问题，使用时间8年
- 操作系统
 - 熟悉Linux系统，对Linux内核的内存管理、进程管理、文件系统、网络协议栈有一定了解，使用时间6年

一般结构

- 网络和数据库
 - 熟悉基于Linux POSIX APIs的网络编程，熟悉基于epoll、reactor、多线程、多进程的网络编程模型
 - 熟悉MySQL的存储原理、innodb存储引擎、索引、binlog同步，并能解决工作中的实际问题
- 其它技能
 - Redis、ZeroMQ、Zookeeper、Etcd、Protobuf、RPC等等，一项一条

参考范例-1

熟练掌握C/C++语言的使用技巧，代码规范整洁，具有不错的代码品味；
阅读过《深入理解计算机系统》和《UNIX环境高级编程》，计算机基础扎实；
阅读过Redis源码，了解Redis的存储模型和基本原理；
阅读过Nginx源码，了解Nginx的网络模型和基本原理，能用Nginx做二次开发；
掌握ZeroMQ、libevent等网络开源库的使用，熟悉网络编程；
熟悉TCP/IP、HTTP和MQTT协议，封装过HTTP协议，实现过MQTT协议部分功能；
能独立承担中小规模项目、模块、功能的设计、开发、调试及维护工作；
有较强的学习能力和自我驱动力，有较强的分析问题和解决问题的能力。

参考范例-2

熟悉C/C++开发，熟悉面向对象编程思想，熟悉常用的设计模式，熟悉STL；
熟悉Windows、Linux和UNIX操作系统，了解线程、进程、IPC及内存管理；
掌握Socket网络编程，熟悉TCP/IP等网络协议；
熟悉select、epoll等多路I/O复用模型；
熟悉常用的数据结构，以及基本的查找和排序算法；
熟悉MySQL、Redis和MongoDB的基本原理，并能在项目中熟练使用；
熟悉Linux和UNIX环境下的常用命令，了解Makefile和gdb，掌握正则表达式；
熟悉Git、SNV等工具的使用。

参考范例-3

熟练掌握C/C++语言，熟悉C++11，四年使用经验；

熟练掌握Go语言，三年使用经验；

熟悉MySQL和Redis的原理，能够在项目中熟练使用并进行性能优化和参数调优；

熟悉Kafka、RabbitMQ等主流消息中间件，能够在项目中熟练使用；

熟悉Linux系统的基本原理和网络协议，能够进行性能优化和参数调优；

熟悉TFS、GFS和Ceph等分布式存储系统的原理；

熟悉Zookeeper、Etcd等分布式协同系统的原理，能基于Etcd开发服务注册中心。

参考范例-4

掌握计算机网络、操作系统、软件工程的基础知识；
熟悉C/C++语言，理解面向对象编程思想，熟悉基于C++11的多线程编程；
熟悉Linux常用命令，熟悉Linux下gcc、gdb、cmake和make等开发工具的使用；
熟悉Linux中的多线程编程和并发网络服务，熟悉基于select、poll和epoll等多路I/O复用技术的高并发编程模型；
熟悉网络七层模型和TCP/IP协议栈的层次结构，掌握TCP、UDP、HTTP、RTSP、RTCP、RTMP、HLS、HTTP-FLV和WebRTC等常见网络协议和流媒体协议，熟悉TCP的三路握手和四次挥手，了解TCP的流量控制和拥塞控制；
掌握数字签名原理，了解HTTPS安全机制的工作原理；
了解Nginx、OpenResty、MySQL、Redis、ZeroMQ和Zookeeper等常用网络框架、数据库和中间件的使用。

两点建议

- 通过专业技能能看清技术栈和技术实力
- 专业技能七八条，每条三四个专业名词

项目经验

- 基本原则
- 参考范例
- 常见误区

基本原则

- 项目不要太多，两三个，最多四个
- 针对目标岗位的技能要求选择项目
- 项目背景、产品功能不要写，凸显技术实现，淡化业务需求
- 工程师的职责是在规定的时间内使产品落地，核心算法不是工程师关注的重点
- 项目中不要写研究性的东西，不确定的东西，要有确切结果实实在在的东西
- 简历写得越具体，面试准备越充分

参考范例-1

项目名称： IM即时通信系统

项目概述： 一款可用于企业内部的即时通信软件系统平台，提供PC端一对一聊天、群聊、文件共享、图片及音视频传输、头像及表情、消息数据持久化等功能。系统后端服务架构主要由LoginServer、MsgServer、RouteServer、DBproxyServer、FileServer和MsfsServer这六个服务器组成。另外通过WebServer和HttpMsgServer实现后台对用户的注册和管理。

开发环境： Linux、Windows、C/C++、Nginx、Redis、MySQL

参考范例-1

使用技术： Nginx、DNS域名解析协议、HTTP/HTTPS、WebSocket、PHP、基础组件封装、epoll封装、TCP/IP协议栈、ThreadPool和BufferPool、reactor反应堆、ZeroMQ消息队列中间件、OpenSSL开源加密库、JSON、编码和解码、压缩和解压缩、Log日志系统、Redis/MySQL连接池、数据库访问、基于P2P网络的去中心化技术、CMake和Makefile、配置数据和文件读取。

参考范例-1

本人职责：

- 1.研究基于P2P网络的去中心化技术并实现整体架构；
- 2.通过Nginx实现MsgServer的水平扩展和负载均衡；
- 3.基础组件ThreadPool、BufferPool和netlib的开发和封装；
- 4.通过Redis/MySQL连接池将消息数据持久化到数据库；
- 5.通过自定义协议实现服务器间的数据传输，利用ProtoBuf实现二进制数据流的序列化及反序列化；
- 6.利用ZeroMQ消息队列中间件提高MsgServer和DBproxyServer间的数据吞吐量。

参考范例-2

项目名称： 网课直播平台

项目概述： 网课直播平台将推流拉流从后台服务器业务系统中分离出来，通过SRS流媒体服务器实现推流拉流，后台服务器业务系统的业务包括用户注册与登录、直播列表和房间逻辑管理、用户权限和系统设置、日志收集等。

开发环境： Linux、Windows、C/C++、Nginx、MySQL

参考范例-2

使用技术： Nginx、线程池、内存池、网络库的封装、工厂模式、策略模式、STL、DNS协议、TCP协议、RTMP应用层协议、FFmpeg框架、AAC音频编解码、H.264视频编解码、FFplay播放视频流、AES加密算法、MySQL数据库访问。

本人职责：

- 1.设计并实现直播推流端、观众拉流端和业务服务端；
- 2.实现RTMP应用层协议、搭建SRS流媒体服务器；
- 3.基于FFmpeg框架的流媒体编解码、实现推流拉流、研究H.264视频编码标准；
- 4.解决播放端音画同步问题；
- 5.通过对缓冲区的优化管理，解决因花屏、卡顿、延迟等导致的体验度差的问题。

常见误区

- 工作中的重点和项目中的重点不是一回事
- 选一个合适的开源工具解决一个问题，也就是半天的事，比花半个月改掉一个Bug有含金量得多。简历中值得大书特书的东西往往并不是花时间最多东西
- 在项目中花时间最多的往往都是业务问题，这对下一家公司其实没什么意义，相反一个简单技术的使用，恰恰可能使新公司从中受益，更值得放到简历中

自我评价

- 不要自我评价
- 参考范例
- 建议写法

不要自我评价

- 自我评价不是自己评价自己，更不是自己夸自己
- 自我评价不是自己介绍自己，平淡无奇毫无亮点
- 自我评价的艺术在于只讲事实，不判断，不评价，让看简历的人去评价
- 利用自我评价弥补技术空白和经验短板

参考范例-1

阅读Nginx线程池、内存池源码，了解其工作机制；

阅读ZeroMQ源码，了解其高性能的内部工作机制、区分大小消息、零拷贝、批量接收和发送消息、读写线程的无锁算法实现、批量预读机制等；

阅读开源协程源码，了解协程的上下文切换和调度算法的实现，以及用同步编程实现异步性能的技巧；

阅读有关C++对象模型的书籍。

参考范例-2

喜欢研究高并发、高性能服务器的设计，C10K、C1000K问题研究；
喜欢研究各种开源软件，如Nginx、Redis、Zookeeper、LVS、Keepalived、libevent、libco等；
喜欢研究分布式架构设计、分布式文件系统等；
喜欢研究DPDK、TCP/IP，自己动手实现了一个用户态的TCP/IP协议栈；
喜欢研究Linux内核相关的功能与特性。

参考范例-3

热衷于跟踪前沿技术，撰写技术文章：

GitHub: <https://github.com/sunhan3787>

博客: <https://www.cnblogs.com/will287248100>

沉迷于源码阅读，如Nginx、Redis、FFmpeg等。

参考范例-4

利用业余时间编写过一个基本稳定的底层软件框架，包括网络通信层(epoll+socket)、多线程任务队列、线程池、数据库访问、基于生产者消费者模型的消息队列。基于该框架实现了一个IM即时通信服务器。该框架还被用于某公司的期货行情、交易C++服务器，现已成功上线，运行稳定。

建议写法

- 阅读书籍，年阅读量
- 参与开源项目，阅读开源代码
- 技术博客，共多少篇，周更多少篇
- 参加开源大会，线上学习课程
- 抗压能力强，能够接受996，必要的时候007

荣誉证书

- 加分项，没有荣誉证书可无此项

大道至简

- 为了拿到好Offer，认真准备一份好简历即成功了一半
 - 好简历既是展示自己的机会，也是一份对对方的尊重和感激
 - 好简历既能凸显自身的优势，也能弥补自身的劣势和偏弱项
 - 好简历可以为面试指明方向，在面试过程中为自己争取主动
 - 好简历可以为面试增加谈资，不要在尬聊中与机会失之交臂
 - 好简历不乏专业性技术用语，平时多积累多读书多看源代码
 - 好简历从来无惧技术性难点，知之为知之不知为不知是知也
- 长风破浪会有时，直挂云帆济沧海
 - 预祝大家都能找到好工作，开启精彩的职业人生