

# 20180403阿里c/c++研发工程师实习面试

## 1. 自我介绍

- 1) 自己的简单情况：姓名，年龄，毕业院校，专业，兴趣爱好、性格特点等
- 2) 优点与长处：技能、获奖、专业知识、学术背景等
- 3) 对应聘职位的想法和规划：对该职位的理解、自己具备什么条件来做好这份工作
- 4) 结束语：我希望得到这份工作，我一定会尽自己所能为企业的发展做出贡献

## 2. 根据自我介绍提一些问题

2.1 介绍一个自己写的c/c++小程序，说了个动态规划的，提问动态规划的实际应用场景

答：布隆过滤器

3.之后根据简历来提问 3.1 linux下网络编程的 3.1.1 tcp的连接建立，故障中断处理，结束方式； 答：tcp建立连接需三次握手(

第一次：client-> SYN=1 seq=x ->server; client<- SYN=1 ACK=1 seq=y ack=x+1 <- server; client-> ACK=1 seq=x+1 ack=y+1 ) tcp释放连接需四次挥手(

第一次：

client -> FIN=1 seq=x -> server client <- ACK=1 seq=y ack=x+1 <- server client <- FIN=1 ACK=1 seq=z <- server client -> ACK=1 seq=u ack=z+1 -> server )

故障中断处理(如：客户主机崩溃、客户进程崩溃、网络连接断开等) signal(SIGPIPE, SIG\_IGN); //服务器崩溃响应一个RST给客户进程，客户进程向收到RST的套接字执行写操作时，

内核发送SIGPIPE信号 **signal(SIGCHLD, sig\_chld);**//服务器处理僵死进程

3.1.2 udp发送数据的优化方式（参数设置之类）；

答：setsockopt(sockfd, SOL\_SOCKET, SO\_SNDBUF, 25000); //发送缓冲区增大为25000  
setsockopt(sockfd, SOL\_SOCKET, SO\_SNDLOWAT, ); //发送缓冲区的最小可写位,结合select使用

3.1.3 服务器如何处理多个客户端的方式； 答：三种方式(多进程、多线程、IO复用select)

3.1.4 udp或tcp的实现过程伪代码； 答：

3.2 进程间通信方式 答：管道、信号量、消息队列、信号、共享内存、套接字

3.3 介绍几种服务器，如nginx, tomcat, 阿帕奇 答：**web服务器：Nginx** 是一款轻量级的Web服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器，并在一个BSD-like协议下发行，其特点是占有内存少，并发能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：新浪、网易、腾讯等。

Tomcat 是一个免费的开放源代码的Web应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP程序的首选。

Apache 是世界使用排名第一的Web服务器软件。它可以运行在几乎所有广泛使用的计算机平台上，由于其跨平台和安全性被广泛使用，是最流行的Web服务器端软件之一。

数据库管理系统：MySQL(Linux作为操作系统，Apache和 Nginx作为 Web 服务器，MySQL 作为数据库， **PHP/Perl/Python**作为服务器端脚本解释器。LAMP/LNMP组合), SyBase(SyBase公司), DB2(IBM公司), Oracle(甲骨文公司), SQL Server(微软公司)。

3.4 http协议 答：应用层协议，并没有规定一定使用**Tcp**协议，**HTTP**只假定其下层协议提供可靠传输，一般基于**tcp**，默认端口**80**

4. 简单介绍一下快速排序，堆排序（时间、空间复杂度，是否稳定）

快速排序：  **$O(n \log n)$**  空间：  **$O(\log n)$** , 不稳定

堆排序：  **$O(n \log n)$**  空间 **$O(1)$** ，不稳定

时间复杂度为 **$O(n \log n)$** : 快排，堆排，归并排

空间复杂度：除了快排 **$O(n \log n)$** ，其它都是 **$O(1)$**

稳定有3个：直接插入，冒牌排 和 归并排。

4. mysql慢查找

6. linux命令awk、sed

答： **grep**的查找、**sed**的编辑、**awk**的文本分析

7. 实际编程中出现c/c++内存泄露问题，如何解决的 答：1) 内存泄露是如何造成的?一般指堆内存泄露，应用程序在**malloc/new**等函数分配到一块内存，使用完后，没有显示的用**free/delete**释放该内存，导致这块内存不能被再次使用。2) 检测是否有内存泄露：**valgrind** (需要在编译时增加 **-g**参数)，**ps -au --sort=%cpu** (查看%CPU/%MEM字段)，**top**命令 (查看系统进程情况)，**ls /proc/pid/task | wc -l** (查看进程的线程数)，

**more /proc/pid/status** (查看进程的状态，包括线程数) 3) 定位源代码中导致内存泄露的位置，修改代码，增加**free/delete**函数 8.实习地点意向

总结：根据你简历的或自己实时说的来提问