## **网易一面**

### **一、自我介绍和项目介绍**

简单自我介绍下吧

balabala

你对哪方面是最熟的,是不是Linux C++ 或者说是最感兴趣?

表达自己倾向基础方向开发

问下项目经历吧，能不能详细介绍下这个项目里做了哪些事情，解决了什么问题啊？

详细的介绍了项目和之前准备好的一个问题(socket方向的，自己通过 netstat 等监控工具排查问题经过，讲得很详细，感觉面试官比较满意)

你中间有没有去提升或者改善性能啊，比如说改参数啊或者调整什么机制，有没有这方面的努力？

也是简单介绍了下自己怎么优化的，达到了什么效果(QPS)

蛮多和项目相关的问题就不细说了，反正就是大家准备项目的时候提前准备一些比如遇到的困难、如何解决的、性能如何、如何优化这样子的问题。

### **二、基础**

* 你对 Linux 内核 IO 机制有没有了解，就比如说 select 和 epoll 有什么差别？

哈哈哈问到这个问题我就笑了，然后就开始先说差别，比如select 有文件描述符限制，频繁的内核态和用户态拷贝、O(n)遍历这些，然后讲了 select 和epoll 都是基于文件系统中 file\_operations 的 poll 调用，然后开始详细讲 epoll，大概讲到了内核用红黑树维护监听描述符，就绪队列，回调函数机制避免O(n)的遍历这些，答完面试官就说说得比较全面深入。

* 刚才你说你用 netstat 命令去检查连接状态，你能不能说下这个命令是做什么，看到的链接状态会有哪些？

就讲了ESTABLISHED、SYN\_RCVED、CLOSE\_WAIT、TIME\_WAIT(故意最后答这个引过来)

* 如果是你在部署某一个程序服务的时候，看到大量的TIME\_WAIT连接是怎么回事？

分析了 TIME\_WAIT 是主动关闭一方，所以服务器应该是有大量短连接请求

* 那么大量的 TIME\_WAIT 会导致什么问题呢？

大概从端口占用 2 MSL 不能使用，可能导致后续没有端口接受新连接之类方向分析

* 那我们怎么去避免这个问题，避免大量 TIME\_WAIT

讲了设置 socket 选项 SO\_LINGER 或者设置 SO\_REUSEADDR 让端口和地址重用

* 平时用什么语言，做什么开发？

C++ 一般用来和网络 Linux 相关的开发， Java 的话是偏 Web 后端， Python一般是写脚本或者爬虫也有用后端框架

* Python的话用到过装饰器吗？

嗯，然后讲了装饰器来干嘛，比如无侵入的增加函数功能(比如计时)，有点类似 Java 注解，支持 AOP 编程。乱扯了一堆...

* 那 Java 你有哪些了解得常用设计模式？

讲了装饰器模式和在 jdk IO 流中的运用，单例模式

* 那单例我们一般用来做什么？

只需要一个对象的场景，比如数据库连接池、文件系统这种。。。

* 了解过 Java 里面的线程安全问题吗？

感觉不妙引到 Java 上来了，赶紧给面试官说自己 Java 不是很熟悉，讲下 C++ 中的吧，然后开始 C++ 的问题

* 讲下 C++ 里虚函数

虚函数表 虚函数表指针 函数指针 动态绑定 基类指针指向派生类对象这些，还有接口设计方面

* 在 C++ 里面结构体和类的区别

默认访问权限 + 一点其它自己的使用感觉

* 用过智能指针吗，能不能介绍下它解决的问题和你使用的经验？

首先讲了裸指针生命周期管理的困难，内存泄漏、野指针这些

* 然后讲了智能指针是如何利用 RAII 来解决的，然后顺便提了那几个智能指针区别

讲了自己项目中如何利用 shared\_ptr 的，还有如何用 weak\_ptr 解决环形引用的

* 刚才你提到内存泄漏，那你是如何在 debug 发现或者是定位内存泄漏的呢？

讲了下 Valgrind 套件下的 memcheck

* 那后面用了智能指针之后你再去跑内存泄漏有减少吗？

嗯，工具检测没有内存泄漏发生了

* 可以讲下树的深度遍历和广度遍历

前 中 后  用栈或者递归

层次  用队列

* 讲一下 hash 表

详细讲了下 STL unordered\_map 底层

* 那哈希表冲突怎么解决？

线性探测和平方探测，说了这两个有主聚集和次聚集缺点

大多数语言 map 用的是开链法

* 如何判断链表是否有环？

说了用 set 和快慢指针两种

* OK，我问几个操作系统方面的问题哈

### **三、OS**

* 你知道 copy on write 吗

详细讲了 fork 的COW, 锁定父子进程页表只读，任何一进程写时就分配页框复制之类，顺便还提了下 C++ 中 COW 在string 上的应用

* 实现进程间通信的方式

匿名管道 有名管道 共享内存 信号量 消息队列 socket， 然后强调自己偏向使用 socket 通信，因为方便将单机多进程扩展到多机多进程

* 你单机开发和实验用的什么 Linux 开发版啊？

讲了 Ubuntu 和 deepin

* 了解一些运维命令吗，比如你在测试的时候怎么看负载啊

答了 top

* 后面详细问了 top, 比如问我知道 Load Avg 代表什么含义，进程的 CPU 利用率会超过 100% 吗，top 还能看什么性能指标？
* 你有什么问题问我吗

## **面经结束**

### **彩蛋开始**

说实话，大家看完这份面经，是不是觉得也没问什么特别难的问题？

但是这个面试我印象很深刻，因为还有一些问题没记下来，印象里就是面试官一直追问，然后各自问题轮番上阵。

从网络编程到 TCP 状态、从 C++ 虚机制到多线程、COW 这些话题，还有 Python、Java、C++ 这些语言都会问，跨度也很大。

但是我基本都能回答得上来，并且会刻意引导到我熟悉的方向。

比如问 TCP 状态的时候，我会刻意将 TIME\_WAIT 回答到最后一个，然后果然面试官就顺着我，问部署服务是否遇到过大量这个状态之类....

比如我当时对 Java 多线程那块不是特别熟悉，面试官问我的时候，我直接让他问我 C++ 的，给讲 C++ 里多线程怎么做的。

这个面试官对我评价也很不错，后来网易二面这个面试官，也就没问我这个基础问题了，说一面问了很多，我的基础也很不错。

然后直接跟我聊云计算、K8s这些，但是恰好，虚拟化这块我了解不多，仅仅对 Docker 的原理有一些粗浅的认识。

然后后面就建议我不要去他们那暑期实习，两个月的时候不太够，让我明确自己到底想做什么（传说中的好人卡？不过我感觉不至于，这个面试官是真的出于对候选人负责。

和这个面试官聊了一些无关面试的话题，然后我就决定去腾讯实习了，拒掉了后面头条这些约的面试。

话说回来，刚才说了面试要引导到自己熟悉的方向，能引导的前提就是你了解得足够多，不然你怎么知道该往哪引导呢？

其次就是知识要系统，比如 Copy On Write 这个话题，能从 fork 联想到 C++ 中的应用。

另外就是知识面要尽可能宽一些，我以前是很爱看各种技术文档、博客的，所以了解也很多。

还有就是编程语言，我一直说的就是，有时间的话，C -> C++/Java -> Python 都好好学一下。

并且有了 C/C++ 基础，后面两门语言都很快的，而且你很快能掌握 Python 这些底层实现机制，如果你对 C++ 对象模型这一套了解的话，这些都是换汤不换药的。

在不涉及 Java 那些 Spring 全家桶的前提下，光是语言还是比较好学。

最后就是基础知识真的要好好刷书、刷视频，不要直接看面经，今天晚上我和一个同学电话了近 20 分钟，他讲了下他目前的学习路线：

基本是刷题 + 刷面经，并且是 数据结构、计算机网络、操作系统这些都没有系统的过一遍，这样肯定是不行的啊。

面经刷得多，可能面试也能过，但是很难系统化，面经上没有刷的点，你就漏掉了。

所以我给他的建议就是：

数据结构 + 计网 + 操作系统 都采用 刷视频 + 刷书的方式系统学习一遍，并且绕过非重点内容。

系统学习后，再去看面经，提炼高频问题。

到底是知识形成系统？