副本 面试总结

前期面试准备

算法

基础知识

项目经验

场景分析

1. 整体流程

一般三轮技术面和一轮hr面试

第一轮：

一般是你同事或者比你级别稍微高一点的。

主要问你一些基础或者项目。同时会让你**写个算法题**

第二轮：

一般就是你后期的leader

会问你一些方案和项目，也会有一部分的基础。

会让你写算法题。

各种解决方案。（偏技术）

第三轮：

项目部的技术负责人或者CTO这类的。

会问你项目，各种解决方案。

第四轮：

一般hr面试

会问你工作经历

为啥出来看机会

薪资要求

选择我们的原因是啥？ 一定要表达出强烈想去他们公司的意图，就是死命舔

2. 自我介绍

每次都会让你做自我介绍，可以从下面角度说

可以从公司是干什么的，

你是哪个部门的，

你们这个部门是做什么的

你负责啥?

擅长那些技能

负责过那些业务难或者技术难的业务

你的性格。

平日是否爱学习、爱思考（有证据）

是否有自己博客、git或者项目

3. 算法

大厂面试主要喜欢考察**动态规划、链表、字符串排序**。

难度一般**简单到中等**之间。

代码行数在10-30行之间(如果解法太长的题目优先级降低)。

1.1 刷题推荐

1. 力扣网前100道简单和中等难度的

   https://leetcode-cn.com/problemset/all/

2. 牛客网高频200道题中的前100道 （前30道简单和中等的必刷）

   https://www.nowcoder.com/ta/job-code-high

力扣题解推荐：

https://leetcode.wang/

https://labuladong.gitee.io/algo/

1.2 突击题目

1. 数组

1. 两数之和 https://leetcode-cn.com/problems/two-sum/

11. 盛最多水的容器 https://leetcode-cn.com/problems/container-with-most-water/
15. 三数之和 https://leetcode-cn.com/problems/3sum/
26. 删除有序数组中的重复项 https://leetcode-cn.com/problems/remove-duplicates-from-sorted-array/
31. 下一个排列 https://leetcode-cn.com/problems/next-permutation/
33. 搜索旋转排序数组 https://leetcode-cn.com/problems/search-in-rotated-sorted-array/
283. 移动零 https://leetcode-cn.com/problems/move-zeroes/

2. 链表

2. 两数相加 https://leetcode-cn.com/problems/add-two-numbers/
19. 删除链表的倒数第 N 个结点 https://leetcode-cn.com/problems/remove-nth-node-from-end-of-list/
21. 合并两个有序链表 https://leetcode-cn.com/problems/merge-two-sorted-lists/
23. 合并K个升序链表 https://leetcode-cn.com/problems/merge-k-sorted-lists/
24. 两两交换链表中的节点 https://leetcode-cn.com/problems/swap-nodes-in-pairs/
141. 环形链表 https://leetcode-cn.com/problems/linked-list-cycle/
24. 两两交换链表中的节点 https://leetcode-cn.com/problems/linked-list-cycle-ii/
206. 反转链表 https://leetcode-cn.com/problems/reverse-linked-list/

3. 树

二叉树的前中后遍历 https://leetcode-cn.com/problems/binary-tree-inorder-traversal/
二叉树的层序遍历 https://leetcode-cn.com/problems/binary-tree-level-order-traversal/
110. 平衡二叉树 https://leetcode-cn.com/problems/balanced-binary-tree/
101. 对称二叉树 https://leetcode-cn.com/problems/symmetric-tree/
剑指 Offer 54. 二叉搜索树的第k大节点 https://leetcode-cn.com/problems/er-cha-sou-suo-shu-de-di-kda-jie-dian-lcof/

4. 动态规划

42. 接雨水 https://leetcode-cn.com/problems/trapping-rain-water/
53. 最大子序和 https://leetcode-cn.com/problems/maximum-subarray/
152. 乘积最大子数组 https://leetcode-cn.com/problems/maximum-product-subarray/
64. 最小路径和 https://leetcode-cn.com/problems/minimum-path-sum/
300. 最长递增子序列 https://leetcode-cn.com/problems/maximum-product-subarray/
70. 爬楼梯 https://leetcode-cn.com/problems/climbing-stairs/
121. 买卖股票的最佳时机 https://leetcode-cn.com/problems/best-time-to-buy-and-sell-stock/
122. 买卖股票的最佳时机 II https://leetcode-cn.com/problems/best-time-to-buy-and-sell-stock-ii/
198. 打家劫舍 https://leetcode-cn.com/problems/house-robber/
213. 打家劫舍 II https://leetcode-cn.com/problems/house-robber-ii/
337. 打家劫舍 III https://leetcode-cn.com/problems/house-robber-iii/
300. 最长递增子序列https://leetcode-cn.com/problems/longest-increasing-subsequence/

322. 零钱兑换 https://leetcode-cn.com/problems/coin-change/

5. 其他

排序 https://www.nowcoder.com/practice/2baf799ea0594abd974d37139de27896

设计LRU缓存结构 https://www.nowcoder.com/practice/e3769a5f49894d49b871c09cadd13a61

二叉树先序，中序和后序 https://www.nowcoder.com/practice/a9fec6c46a684ad5a3abd4e365a9d362

合并类题目：合并有序数组、链表、多个链表或者数组的。并排序

另外必看：冒泡排序和快速排序

4. 基础知识

主要有下面几大块

core java

重载和重写的区别

接口和抽象的区别，项目中是怎么用的?

Exception和Error

Object类内的方法

Boolean占几个字节

进程和线程的区别

JUC（必问）

必问，一般是一面问的多

1. volatile 的用处和实现?

保证可见性　如何实现?

不保原子性　　如何实现?

部分保证有序性　如何实现?

lock锁

2. synchronized（重点，必考）

锁的对象?

java对象头结构

不同java版本对其优化?

无锁、偏向锁、轻量级、重量级锁都是怎么晋升的，对象头里是怎么变化的

synchronized和Lock的区别（必考）

3. CAS

cas是啥？为啥要有?

底层实现

CAS实现原子操作的三大问题？解决方案

Atomic类如何保证原子性（CAS操作）（必考）

4. 指令重排序

有哪几种（问的不多）

5. 线程池（重点，必考）

线程池的7个参数，必须精通其内部含义（必考）

ThreadPoolExecutor的工作流程（必考）

线程池的状态生命周期，他们是怎么流转的

运行(RUNNING) 关机(SHUTDOWN) 停止(STOP) 清理(TIDYING)终止(TERMINATED)

你们项目中怎么用的线程池

6. 锁

让你实现一个死锁

让你实现一个单例，加锁（synchronized和lock 2种版本）

什么是公平锁和非公平锁

7. AQS

AQS是啥?

内部是怎么工作的?

让你实现一个AQS，你准备怎么设计?

里面的有几个属性，都是什么意思

8. 集合重点（重点，必考）

1. HashMap和ConcurrentHashMap区别（必考）

2. ConcurrentHashMap的数据结构（必考）

CopyOnWriteArrayList

底层jdk1.7 1.8 都是怎么实现的 。是用锁还是cas。数组和红黑树的晋升条件

9. ThreadLocal

内部结构是啥？原理和实现

什么是内存泄露？产生原因、避免措施

你们项目是怎么用的?

父子线程传参怎么办?

网络

这个版块问的不多

基础网络部分

1. udp和tcp的区别
2. 三次握手和四次挥手
3. tcp头的结构
4. http头的结构
5. https的接口
6. cdn是什么东西。
7. 什么是前后端分离，你们项目中怎么实现的。
8. 讲述一下一次完整的网络请求是什么步骤

IO

bio nio aio是啥

底层是怎么实现的

JVM

1. 运行时数据区域（内存模型）（必考）堆栈，共有还是私有的

2. 垃圾回收机制（必考）

3. 垃圾回收算法（必考）

4. 各垃圾回收器的特点及区别 CMS或者G1的回收过程，优缺点，**会问的非常多**

5. 类的加载过程

6. 内存泄露了怎么办？你们是怎么解决的

7. JDBC和双亲委派模型关系，怎么破坏

8. 问一些jvm参数 xms或者jstat的一些

Spring

spring很怪 问的很少，源码也不会问，大概率是面试官也背不过？ 主要问DI和AOP

1. Spring的IOC/AOP的实现（必考）

2. 动态代理的实现方式（必考）

3. AOP？是啥 底层实现。 cglib的底层实现 （必问题目）

4. Spring如何解决循环依赖（三级缓存）（必考）

5. Spring的后置处理器

6. Spring的@Transactional如何实现的（必考）

7. Spring的事务传播级别

8. BeanFactory和ApplicationContext的联系和区别

9. springboot的是怎么工作的，和spring的区别

Mysql（必问）

1. 事务的基本要素

2. 事务隔离级别（必考）

3. 如何解决事务的并发问题(脏读，幻读)（必考）

4. MVCC多版本并发控制，底层实现。（必考）

5. binlog,redolog,undolog都是什么，起什么作用

6. 给定隔离级别下，加锁的过程，表锁、行锁、间隙锁、共享锁、排它锁

7. innodb和myisam的区别和优缺点

8. 为什么选择B+树作为索引结构（必考）

9. 索引B+树的叶子节点都可以存哪些东西（必考）

10. 查询在什么时候不走（预期中的）索引（必考）

11. sql如何优化

12. sql的执行顺序？或者 order by原理

13 让你写个sql，一般是join子查询的居多（必考）

Redis（必问）

1. redis的底层数据结构是啥，晋升条件，最大容量是啥是啥。

2. 为啥这么快，应用场景

3. 他是单线程的吗？单进程的吗？

4. Redis的持久化机制 ，参数、流程（必考）

5. 主从同步是怎么做的？同步失败了怎么办?还是失败怎么办

6. 热点key、大key是怎么发现的和解决的

7.缓存穿透、缓存击穿、缓存失效 都是啥

8. 淘汰策略都有啥。内存满了还处理请求吗（必考）

9. 了解bitmap 布隆表达式 lua吗

10 分布式锁是怎么实现的？续期和安全性是怎么保证的

11 他和mencache的区别

12 集群或者哨兵模式下是怎么工作的

13 怎么保证原子性。

Kafka

1. kafka的整体架构是啥，都有哪些部分促成的，作用是啥

2. 怎么保证消息的不丢失。

3. 怎么保证消息的有序性

4. 怎么提高kafka的并发，生产和消费都说一下

5. 为啥kafka这么快

5. 项目

1. 项目里职责

1. 会问你一个部门有几个人，组成状态

2. 这个项目多少人。

3. 这个项目你负责了啥，占了啥地位或者比重，工期啥的

2. 项目细节

1. 让你介绍一下项目整体流程

   一定要有套路，先说架构或者整体，再细说细节

2. 会问你难点和亮点，怎么优化或者解决的

   并发大?

   技术难度大?

   bug或者坑太多?

   你是怎么发现的或者解决的?

如果你实际没用过，但是知道如果用了哪个技术栈会提高并发啥的，可以尽管直接说，还可以怎么怎么优化就行。

==========

核心思路:

面试官其实不是想问你项目curd，**而是你在项目里面的亮点和难点**

你需要让面试官认为你这个项目很难，技术挑战很高，然后你是怎么克服的。

3. 项目实践

如果你项目实在平平无奇，那么你可以融入下面几个亮点

1. 分布式ID是啥?

分布式ID一般是解决数据库主键自增的。

或者解决异步落库订单重复

雪花ID，redis自增，自制ID（时间+外键+随机 拼接的），分布式ID中间件（tiID）。

ID单个获取还是分批获取

2. 分布式事务有哪几种?

你们怎么用的,会问你

XA TCC

3. 分布式锁

redis锁，zk

是怎么续约的

是怎么保证安全性的

优化点：内存锁+分布式锁，用来降低分布式锁的压力和竞争。

4. 限流器都有哪些实现方式

内存 guava的RateLimiter

redis 自增ID，或者lua脚本

快速失败还是超时失败 推荐快速

5. 多级缓存

内存缓存+外置缓存

内存缓存 ： guavaCache Ehcache Caffeine

外置缓存：redis memcache

你们缓存是怎么更新的：永不失效，异步加锁更新

6. 内存泄露怎么解决的。

线程、cpu、内存多方面考虑