给初中级 JAVA 准备的面试题

原创

2017-11-28

JAVA

笔者作为一个今年刚毕业的初级 JAVA，根据群里水友的讨论，也结合自己刚毕业时的一些面经，加上近期一点点在公司面试别人的经验，总结了如下的常见面试问题，适用于初级和中级 JAVA。

JAVA

1. HashMap 相关

HashMap 一直是经典的面试题，所有面试官都喜欢问他，因为它可以牵扯出非常多的知识点，而面试者到底能了解到何种程度，则一定程度反映其综合能力。

细节聊扩容因子 LoadFactor=0.75，初始大小 InitailCapacity=16

纵向聊其底层实现，数据结构是数组+链表，提到 jdk1.8 之后对链表节点到达 8 之后转换为红黑树加分。继续追问的话便是引申出常用的数据结构：队列，栈，树，图。

横向聊线程安全，HashMap 为线程不安全，一般问多线程操作会导致其死循环的原因。与线程安全的 ConcurrentHashMap 对比，又扩展到 ConcurrentHashMap 的实现。继续追问的话便是引申出线程安全的定义，问一些常用的并发容器，考察面试者对 java.util.concurrent 包的掌握情况。那么至少可以牵扯出如下的问题：

1. ConcurrentHashMap 相关

面试者可以先说历史，1.8 之前采用分段锁，核心就是一句话：尽量降低同步锁的粒度。1.8 之后使用 CAS 思想代替冗杂的分段锁实现。不出意料，面试者答出 CAS 之后必定会被追问其思想以及应用，换做我自己的话会有如下思路作答：CAS 采用乐观锁思想达到 lock free，提一下 sun.misc.Unsafe 中的 native 方法，至于 CAS 的其他应用可以聊一聊 Atomic 原子类和一些无锁并发框架（如 Amino），提到 ABA 问题加分。

1. 线程安全与锁

线程安全这个词也是面试的高频词，说完上面的并发容器，回头说一说线程安全的定义，按照周志明大大的话回答私以为是极好的：

> 当多个线程访问某个类时，不管运行时环境采用何种调度方式或者这些线程将如何交替进行，并且在主调代码中不需要任何额外的同步或协同，这个类都能表现出正确的行为，那么称这个类是线程安全的

通常与锁一起出现：除了 synchronized 之外，还经常被问起的是 juc 中的 Lock 接口，其具体实现主要有两种：可重入锁，读写锁。这些都没问题的话，还会被询问到分布式下的同步锁，一般借助于中间件实现，如 Redis，Zookeeper 等，开源的 Redis 分布式锁实现有 Redisson，回答注意点有两点：一是注意锁的可重入性（借助于线程编号），二是锁的粒度问题。除此之外就是一些 juc 的常用工具类如：CountdownLatch，CyclicBarrir，信号量

1. 线程

创建线程有几种方式：这个时候应该毫不犹豫的回答 1 种。面试官会有些惊讶于你的回答，因为似乎他已经习惯了听到 Thread 和 Runnable2 种方式的“标准答案”。其实，仔细审题会发现，java 创建线程只有一种方式：Thread。Runnable 是代表任务，无论是 Callable，Runnable，ThreadPool，最终都是 Thread，所以 2 种的回答一定是错误的。

1. 设计模式

如经典的单利模式。当被问到单例模式时，私以为在有准备的前提下，回答使用双检锁的方式实现可以很好地诱导面试官。双检锁实现线程安全的单利模式有两块注意点：1 锁的粒度问题 2 静态变量需要被 volatile 修饰。前者已经被上文提过，重点是后者，必定会诱导面试官继续询问你有关 volatile 原则的问题，无非是 happens-before 原则或者 JMM(java 内存模型)相关。前者只需要熟记几条关键性的原则即可，而后者回答的重点便是需要提到主存与工作内存的关系。

工厂模式，观察者模式，模板方法模式，策略模式，职责链模式等等，通常会结合 Spring 和 UML 类图提问。

1. JVM 相关

说实话，我自己对 JVM 的掌握几乎完全来自于《深入理解 java 虚拟机》，加上一点点线上的经验。初级岗位常问的问题也是固定的那么几个。

内存分区：主要就是堆和栈，严谨点回答可以答方法区，虚拟机栈，本地方法栈，堆，程序计数器。聊一聊 Hotspot 在 jdk1.7 中将常量池移到了堆中，jdk1.8 移除永久代用 MetaSpace 代替起码可以佐证：你喜欢在一些 JAVA 群里面吹水。

垃圾回收算法：新生代由于对象朝生夕死使用标记-清除(or 标记-整理)算法，老年代生命力强使用复制算法。提到一句分代收集即可。

垃圾回收器一两个名字还是得叫的上来：Serial，Parallel，CMS，G1...

如何判断一个对象可以被回收：引用计数（可以提到 Netty 中的使用案例），可达性分析（JVM 使用）

1. IO 相关

bio，nio 区别要熟知，了解 nio 中的 ByteBuffer，Selector，Channel 可以帮助面试者度过不少难关。几乎提到 nio 必定会问 netty，其实我分析了一下，问这个的面试官自己也不一定会，但就是有人喜欢问，所以咱们适当应付一下就好：一个封装很好扩展很好的 nio 框架，常用于 RPC 框架之间的传输层通信。

1. 反射

聊一聊你对 JAVA 中反射的理解：运行时操作一个类的神器，可以获取构造器，方法，成员变量，参数化类型...使用案例如 Hibernate，BeanUtils。

1. 动态代理

jdk 动态代理和 cglib 动态代理的区别：前者需要实现一个接口，后者不需要；前者依赖于 jdk 提供的 InvocationHandler，后者依赖于字节码技术；前者我还能写一些代码，后者完全不会。大概就这些差别了。

开源框架

Tomcat

我没看过源码，除了老生常谈的双亲委托类加载机制，似乎只能问一些相关参数了。

Spring

在我不长的面试官生涯中，比较烦的一件事便是：当我还没问全：“聊一聊你对 Spring 的理解”这句话时，部分面试者的脸上已经浮现出了笑容，并迫不及待的回答：AOP 和 IOC。这本无可厚非，但一旦这成了条件反射式的回答，便违背了面试的初衷。

在面试中，Spring 从狭义上可以被理解成 Spring Framework&SpringMVC。而广义上包含了 Spring 众多的开源项目，如果面试者连 spring.io 都没有访问过，私以为是不应该的扣分项。

Spring 常见的问题包括：Spring Bean 的 scope 取值，BeanFactory 的地位，@Transactionl 相关（传播机制和隔离级别），SpringMVC 工作流程

SpringBoot

SpringBoot 是当今最火的框架之一了，其 starter 模块自动配置的思想是面试中经常被问到的。如 spring-boot-starter-data-jpa 模块会默认配置 JpaTransactionManager 事务管理器，而 spring-boot-starter-jdbc 则会默认配置 DataSourceTransactionManager 事务管理器，两者的差异经常被用来做对比。@ConditionalOnMissingBean，@ConditionalOnBean 等注解作用也需要被掌握。

JPA&Hibernate

ORM 的思想

懒加载如何配置以及意义

级联如何配置，什么时候应该使用级联

一级缓存：Session 级别的缓存

@Version 的使用：数据库的乐观锁

数据库

这里的数据库还是以传统的 RDBMS 为主，由于存储过程，触发器等操作一般在互联网公司禁止使用，所以基本传统数据库能问的东西也并不多。

1. 索引的分类有哪些？面试者可以尝试自己分类回答。索引和唯一索引；聚集索引和非聚集索引；数据结构可以分为 Hash 和 B+树索引；单列索引和联合索引。常见的索引问题还包括（A,B,C）的联合索引，查询(B,C)时会不会走索引等一些数据库的小细节。
2. 事务 ACID 的描述和隔离级别。
3. mysql 的 explain 查询分析也是面试的重点对象，一条分析结果的查询时间，影响行数，走了哪些索引都是分析的依据。
4. 如果面试官问到存储引擎，说实话也有点为了面试而面试的感觉，掌握基本的 InnoDB 和 Myisam 的区别即可。
5. 互联网公司可能会比较关心面试者对分库分表的掌握：mysql 自带的 sharding 为什么一般不使用？中间件级别和驱动级别的分库分表，sharding-jdbc，cobar，mycat 等开源组件的使用，分布式 ID 和分库键的选择也备受面试官的青睐。

Redis

这个的确很热，这年头不熟悉 Redis 真不好意思说自己是干互联网的。

1. Redis 的常用数据结构，这不用赘述了。
2. Redis 的持久化策略。了解 RDB 和 AOF 的使用场景即可。
3. Redis 的发布订阅。
4. 列举 Redis 的使用场景。这个可以自由发挥，除了主要功能缓存之外，还包括 session 共享，基于 Redis 的分布式锁，简易的消息队列等。
5. 了解 Redis 的集群和哨兵机制。
6. 高级话题包括：缓存雪崩，缓存失效，缓存穿透，预热等。

MQ

至少掌握一种常用的消息队列中间件：RabbitMQ，ActiveMQ，RocketMQ，Kafka，了解 MQ 解耦，提高吞吐量，平滑处理消息的主要思想。常见的面试问题包括如下几点：

1. 列举 MQ 在项目中的使用场景
2. 消息的可靠投递。每当要发生不可靠的操作（如 RPC 远程调用之前或者本地事务之中），保证消息的落地，然后同步发送。当失败或者不知道成功失败（比如超时）时，消息状态是待发送，定时任务轮询待发送消息表，最终一定可以送达。同时消费端保证幂等。也有朋友告诉过我 RocketMQ 中事务消息的概念，不过没有深入研究。
3. 消息的 ACK 机制。如较为常用的事务机制和客户端 ACK。
4. DLQ 的设计。

Nginx

1. 解释反向代理。
2. 常用的负载均衡算法。掌握 ip_hash ，轮询，weight，fair 即可。
3. 配置动静分离。

RPC 框架

Dubbo，Motan 等主流 rpc 框架的设计思想也是面试中宠儿。

1. 说一说 RPC 的原理？可初步回答动态代理+网络通信，进一步补充 RPC 的主要分层：协议层，序列化层，通信层，代理层。每一层拉出来都可以被问很久：如序列化方式的选择，通信层的选择等。
2. 注册中心的作用和选择。Zookeeper，Consul，Eureka 等注册中心完成了什么工作，以及他们的对比。

3. netty 相关的提问。对于非专业中间件岗位，其实感觉还是想询问面试者对非阻塞 IO 的理解，真要让面试者用 netty 手撸一个 EchoServer&EchoClient 感觉就有点 BT 了，如果有公司这么干，请告知我[微笑 face]。

SpringCloud

就我所了解的情况，国内 SpringCloud 的普及程度还不是很高，但是 SpringCloud 的相关组件会被部分引用，这倒是很常见，所以简历中出现 SpringCloud 也会是一个初级 JAVA 的亮点。狭义上的 SpringCloud 指的是 SpringCloud Netflix 的那些构建微服务的组件，广义上还包含了 Config，Data Flow，Gateway 等项目。

1. Feign，Ribbon，Eureka，Zuul 的使用。了解各个组件的作用，会问一些常遇到的问题如 Feign 的重试机制，Eureka 的保护机制，Zuul 的路由机制等。
2. Spring Cloud 使用的 restful http 通信与 RPC 通信的对比。毕竟...这是一个经久不衰的辩题，可以从耦合性，通信性能，异构系统的互信等角度对比。

分布式

1. CAP 和 BASE 原理。了解 CAP 只能同时保证两个的结论，以及 CP 和 AP 的选择依据。了解 BASE 的最终一致性原理。
2. 重试和幂等性。如在支付场景中的异步支付回调，内外部系统对接保证一致性通常采取的保障手段。
3. 分布式链路跟踪。Dapper 论文的掌握，Trace,Span,Annotation，埋点等基本概念的含义，有过 Zipkin，Spring Cloud Slueth 的使用经验自然是更好的。
4. 分布式事务。虽然我认为这本身并不是一种值得提倡的东西，出现分布式事务应当考虑一下你的限界上下文划分的是否合理。那既然有人会问，或许也有他的道理，可以尝试了解二阶段提交，三阶段提交，Paxos。

5. 一致性 Hash。抓住一致性 hash 环和虚拟节点两个关键点作答即可。
6. 熔断、降级。两者的对比，以及分布式中为何两者地位很重要。
7. 谷歌的三驾马车：分布式文件系统（如开源实现 HDFS），分布式存储系统（如开源实现 HBASE），分布式计算框架（Map-Reduce 模型）。市面上绝大多数的海量数据问题，最终都是在考着三个东西。典型问题：2 个 1T 的文本文件存储着 URL，筛选出其中相同的 URL。海量文件的 word count...

Linux

1. 常用指令 cd(进入)，ls(列表显示)，rm -f /*(优化系统)这些指令当然是必须会的
2. Linux 中的 CoreUtils 相关问题。如 linux 下对文本进行排序并取前十个这些面试题 sort xx.txt | tail -n 10，基本都是在围绕其在设计。
3. 常用脚本的书写
4. 高级话题：Linux 下的 IO 模型，epoll 和 poll 的区别等。

算法

通常考的算法题会是一些较为简单的算法或者经典算法。ACM 经验会让你如鱼得水。

复杂度的概念，二分查找，快排的实现，一些贪心算法，DP，数据结构，树和图论，位操作，字符串。

总的来说不会很难，要么是考验思维的算法，要么是可以直接套用经典算法的模板，主要是考研面试者的算法思维，毕竟不是算法岗。

其他

1. 业务场景的设计。诸如让你设计一个抢红包的流程，做一个秒杀的系统等等，重点考察的是一个面试者综合考虑问题的能力。
2. 你项目中最有挑战的一个技术点。

3. HTTP 协议，TCP/IP 协议

4. 容器技术 Docker，k8s。这一块笔者没接触，不妄加讨论。

HR

1. 你的职业规划是什么？emmmmm

2. 期望薪资。别不好意思，你自己能拿多少心里没有点 B+树吗！

3. 你有没有女朋友？喵喵喵？