**一、Java相关**

* Arraylist与LinkedList默认空间是多少；  
  ArrayList=10; LinkedList=0
* Arraylist与LinkedList区别与各自的优势List 和 Map 区别；
* 谈谈HashMap，哈希表解决hash冲突的方法；
* 为什么要重写hashcode()和equals()以及他们之间的区别与关系；

重写 hashCode 是为了让 [Java](https://www.baidu.com/s?wd=Java&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao) 中所有使用到 Hash 算法的数据结构能够正常运行，当然如果你保证你的程序中完全不会用到 Hash 算法，那么你也可以不用重写。

* Object的hashcode()是怎么计算的？

根据对象内存地址计算的一个int值。

* 若hashcode方法永远返回1或者一个常量会产生什么结果？
* Java Collections和Arrays的sort方法默认的排序方法是什么；
* 引用计数法与GC Root可达性分析法区别；
* 浅拷贝和深拷贝的区别；
* String s="abc"和String s=new String("abc")区别；  
  new String()会在堆上分配内存。
* HashSet方法里面的hashcode存在哪，如果重写equals不重写hashcode会怎么样？
* 反射的作用与实现原理；
* Java中的回调机制；
* 模板方法模式；
* 开闭原则说一下；
* 发布/订阅使用场景；
* KMP算法（一种改进的字符串匹配算法）；  
  a字符串包含b字符串的话，返回b第一个字符在a中开始的位置。
* JMM里边的原子性、可见性、有序性是如何体现出来的，JMM中内存屏障是什么意思，

**二、多线程**

* AtomicInteger底层实现原理；  
  Unsafe.compareAndSwap();
* synchronized与ReentraLock哪个是公平锁；
* CAS机制会出现什么问题；  
  ABA问题。并发修改次数太多性能还不如重量级锁。

写操作太多似乎不太适合使用。

* 用过并发包下边的哪些类；
* 一个线程连着调用start两次会出现什么情况？  
  正常运行。
* wait方法能不能被重写，wait能不能被中断；  
  不能被重写，可以被中断。
* 线程池的实现？四种线程池？重要参数及原理？任务拒接策略有哪几种？
* 线程状态以及API怎么操作会发生这种转换；
* 常用的避免死锁方法；

**三、JVM**

* Minor GC与Full GC分别在什么时候发生？什么时候触发Full GC;
* GC收集器有哪些？CMS收集器与G1收集器的特点。
* Java在什么时候会出现内存泄漏；
* Java中的大对象如何进行存储；
* rt.jar被什么类加载器加载，什么时间加载；
* 自己写的类被什么加载，什么时间加载；
* 自己写的两个不同的类是被同一个类加载器加载的吗？为什么？
* 为什么新生代内存需要有两个Survivor区？
* 几种常用的内存调试工具：jmap、jstack、jconsole；
* 类加载的五个过程：加载、验证、准备、解析、初始化；
* G1停顿吗，CMS回收步骤，CMS为什么会停顿，停顿时间；
* 栈主要存的数据是什么，堆呢？
* 堆分为哪几块，比如说新生代老生代，那么新生代又分为什么？
* 软引用和弱引用的使用场景（软引用可以实现缓存，弱引用可以用来在回调函数中防止内存泄露）；

**四、数据库**

* 数据库索引，什么是全文索引，全文索引中的倒排索引是什么原理；

全文索引：根据给定的关键词库，把文件中关键词出现的频率和位置记录，按照一定的顺序归纳，然后就建立了一个以关键词为目录的的索引，当查找这个词的时候就可以直接找到这个文件。

* 数据库最佳左前缀原则是什么？
* 数据库的三大范式；
* 左连接、右连接、内连接、外连接、交叉连接、笛卡儿积等；
* 一般情况下数据库宕机了如何进行恢复（什么是Write Ahead Log机制，什么是Double Write机制，什么是Check Point）；
* 什么是redo日志、什么是undo日志；
* 数据库中的隔离性是怎样实现的；原子性、一致性、持久性又是如何实现的；
* 什么是组合索引，组合索引什么时候会失效；
* 关系型数据库和非关系型数据库区别；
* 数据库死锁如何解决；

一般是应用程序导致的，从数据库查看哪个用户锁住，然后查看什么原因锁住。然后再解决问题。

* MySQL并发情况下怎么解决（通过事务、隔离级别、锁）；
* MySQL中的MVCC机制是什么意思，根据具体场景，MVCC是否有问题；
* MySQL数据库的隔离级别，以及如何解决幻读；

**五、缓存服务器**

* Redis中zSet跳跃表问题；
* Redis的set的应用场合？
* Redis高级特性了解吗？
* Redis的pipeline有什么用处？
* Redis集群宕机如何处理，怎么样进行数据的迁移；
* Redis的集群方案；
* Redis原子操作怎么用比较好；
* Redis过期策略是怎么实现的呢？

**六、SSM相关**

* Spring中@Autowired和@Resource注解的区别？
* Spring声明一个 bean 如何对其进行个性化定制；
* MyBatis有什么优势；
* MyBatis如何做事务管理；

**七、操作系统**

* Linux静态链接和动态链接；
* 什么是IO多路复用模型（select、poll、epoll）；
* Linux中的grep管道用处？Linux的常用命令？
* 操作系统中虚拟地址、逻辑地址、线性地址、物理地址的概念及区别；
* 内存的页面置换算法；
* 内存的页面置换算法；
* 进程调度算法，操作系统是如何调度进程的；
* 父子进程、孤儿进程、僵死进程等概念；
* fork进程时的操作；
* kill用法，某个进程杀不掉的原因（僵死进程；进入内核态，忽略kill信号）；
* 系统管理命令（如查看内存使用、网络情况）；
* find命令、awk使用；
* Linux下排查某个死循环的线程；

**八、网络相关**

* 数据链路层是做什么的?
* 数据链路层的流量控制？
* 网络模型的分层、IP和Mac地址在那个层、TCP和HTTP分别在那个层；
* TCP滑动窗口；
* TCP为什么可靠；
* TCP的同传，拆包与组装包是什么意思；
* Https和Http有什么区别；
* Http 为什么是无状态的；
* TCP三次握手，为什么不是三次，为什么不是四次；
* TCP的拥塞控制、流量控制详细说明？
* Http1.0和Http2.0的区别；
* 两个不同ip地址的计算机之间如何通信；
* 地址解析协议ARP；
* OSI七层模型分别对应着五层模型的哪一部分；
* TCP三次握手数据丢失了怎么办？那如果后面又找到了呢？

**九、分布式相关**

* 消息队列使用的场景介绍和作用（应用耦合、异步消息、流量削锋等）；
* 如何解决消息队列丢失消息和重复消费问题；
* Kafka使用过吗，什么是幂等性？怎么保证一致性，持久化怎么做，分区partition的理解，LEO是什么意思，如何保证多个partition之间数据一致性的（ISR机制），为什么Kafka可以这么快（基于磁盘的顺序读写）；
* 异步队列怎么实现；
* 你项目的并发是多少？怎么解决高并发问题？单机情况下Tomcat的并发大概是多少，MySQL的并发大致是多少？
* 什么是C10K问题；  
  过时了
* 高并发情况下怎么办；
* 分布式理论，什么是CAP理论，什么是Base理论，什么是Paxos理论；
* 分布式协议的选举算法；
* 说一下你对微服务的理解，与SOA的区别；  
  数据库的拆分不一样。
* Dubbo的基本原理，RPC，支持哪些通信方式，服务的调用过程；
* Dubbo如果有一个服务挂掉了怎么办；
* 分布式事务，操作两个表不在一个库，如何保证一致性。
* 分布式系统中，每台机器如何产生一个唯一的随机值；
* 系统的量级、pv、uv等；
* 什么是Hash一致性算法？分布式缓存的一致性，服务器如何扩容（哈希环）；
* 正向代理、反向代理；
* 什么是客户端负载均衡策略、什么是服务器端负载均衡策略；
* 如何优化Tomcat，常见的优化方式有哪些；
* Nginx的Master和Worker，Nginx是如何处理请求的；

**十、系统设计相关**

* 如何防止表单重复提交（Token令牌环等方式）；
* 有一个url白名单，需要使用正则表达式进行过滤，但是url量级很大，大概亿级，那么如何优化正则表达式？如何优化亿级的url匹配呢？
* 常见的Nginx负载均衡策略；已有两台Nginx服务器了，倘若这时候再增加一台服务器，采用什么负载均衡算法比较好？
* 扫描二维码登录的过程解析；
* 如何设计一个生成唯一UUID的算法？
* 实现一个负载均衡的算法，服务器资源分配为70%、20%、10%；
* 有三个线程T1 T2 T3，如何保证他们按顺序执行；

1.锁+执行ID(T1=1,T2=2,T3=3),notifyAll(),wait()配合使用。

2.ExecutorService **executor** = **Executors**.*newSingleThreadExecutor*();

3.t1.start();t1.join();t2.start();t2.join();t3.start();

|  |
| --- |
| **t.join();      //使调用线程 t 在此之前执行完毕。 t.join(1000);  //等待 t 线程，等待时间是1000毫秒** |

* 三个线程循环输出ABCABCABC....

**十一、安全相关**

* 什么是XSS攻击，XSS攻击的一般表现形式有哪些？如何防止XSS攻击；