<https://www.nowcoder.com/discuss/239036?type=post&order=time&pos=&page=1>

1.项目介绍

2.int与Integer区别，给数量庞大的数字使用Integer创建是否会对重复数字创建新对象？ 运行时常量池中只会保存范围在-128~127之间的Integer对象。不在这个范围还是会重复创建

3.java垃圾回收算法有哪些，现在商用主要使用哪种回收算法？

既然要回收垃圾就会引出怎么识别垃圾（可达性分析和引用计数法）和怎么回收垃圾（下面三个算法）两个阶段：

垃圾回收算法有：标记复制，标记清除，标记整理。垃圾回收器是对垃圾回收算法的实际应用。（不同的垃圾回收算法对应有不同的对象内存分配策略，指针碰撞和空闲链表）

不同的垃圾收集器使用的算法都是以上3种。可能采取的策略不同以应对不同的场景。

不同的垃圾回收算法适应不同生命周期的对象。所以会有老年代和新生代的区分。对象可能的长短生命周期无法确认，但是有两个准则：也就是下面垃圾回收器分为老年代和新生代的原因：

垃圾回收器分为老年代和新生代的原因：

1. 大多数对象朝生夕死
2. 多次没有被垃圾回收的对象越不容易消失

既然出现了**分代收集**，不可避免会出现**跨带引用**的问题：只需将老年代内存分成若干小块以标识哪块内存存在跨带引用，在新生代中维持一个数据结构记录存在跨带引用的内存块。这样我们就在进行可达性分析（引用计数法）来标记对象时就不用对老年代的对象进行可达性分析，这可以节省很多时间。说到可达性分析，就绕不开**GCroot**，可达性分析就是从GCroot对象来根据引用关系进行可达性分析的。很明显，如果被GCRoot对象引用了，说明这个对象还需要使用，就不能作为垃圾对象被删除。

垃圾收集器：CMS，G1，serial ，serial old，ParNew(新老带采用垃圾回收算法不同，serial和serial old多线程版本), Parallel Scavenge和 Parallel old注重吞吐量的新老年代版本。CMS其一般与ParNew工作。叙述一下CMS 的垃圾回收阶段，G1的垃圾回收阶段。CMS和G1的垃圾回收由于有并发清除阶段，其在清除的同时，用户线程会产生浮动垃圾。由于会产生浮动垃圾，所以其需要有一个触发fullGc 的阈值，以防止垃圾回收速度小于浮动垃圾回收的速度导致的用户进程的内存溢出，只能stop world使用serial old垃圾回收器来处理,停顿时间会延长。

Cms注重停顿时间；G1注重低延迟和吞吐量。G1不按照固定带下的分代区域划分，而是将其分成固定大小的region。但是其回收算法还是有老年代和新生代的分别。每个region再扮演不同的角色。其解决跨带引用还是靠数据结构解决当前region跨带引用的其他region。

G1收集器负责跟踪各个Region里面的垃圾堆积的价值大小，并维护一个优先级列表。优先收集垃圾比较多的region，这是Garbage First的由来。回收多少个region（由用户所希望的停顿时间来制定回收计划，采用标记整理算法（老年代，新生代采用标记整理），其内部维持了一个分配内存的指针来区分已用和空闲，）

1. 算法：二叉树打印所有和为某值的路径。

中序遍历，直接leetcode上

1. 算法：数字中查找第k大。

最小堆，维护一个k容量大小的最小堆。

<https://www.nowcoder.com/discuss/239036?type=post&order=time&pos=&page=1>

**一面：**

自我介绍，自己介绍项目里的技术难点（总是逃不过这个环节，实在是没啥亮点）

2、**redis在项目里的使用，使用的架构（单机版）**

**3、单机版在实际生产中会有什么问题，除了单机版还了解哪些架构方式**

4、问jvm内存模型

(JMM 定义了一套在多线程读写共享数据时（成员变量、数组）时，对数据的可见性、有序性、和原子性的规则和保障;

一些主内存中经常被访问的数据会被缓存到工作内存的高速缓存中，即使主内存更改了数据，线程因为从工作内存读取数据，无法读取到主内存中更改的数据。

注意：主内存是线程共享的，工作内存是线程独占的。

针对上述不可见性问题，可以使用volatile修饰成员变量和静态成员变量。可以避免线程从自己的工作缓存中查找变量的值，必须到主存中获取它的值，线程操作 volatile 变量都是直接操作主存。Volatile只能保证可见性和有序性（禁止指令重排序），不能保证原子性。Sychronized关键字保证可见性和原子性不能保证有序性。

(Volatile靠插入内存屏障防止其后面的指令跑到前面去。底层通过内存屏障防止指令重排，变量前后之间的指令与指令之间有序可见)

JMM关于synchronized的两条规定：(Sychronized是靠操作系统内核互斥锁实现的)

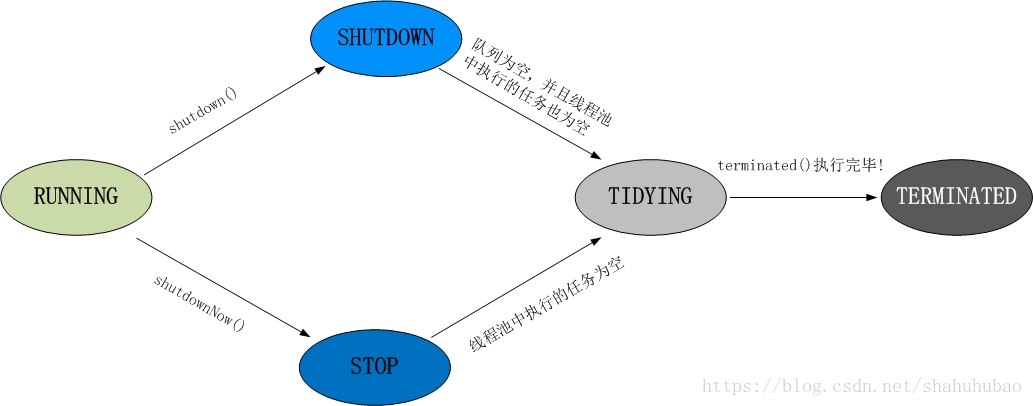
　　1）线程解锁前，必须把共享变量的最新值刷新到主内存中

　　2）线程加锁时，将清空工作内存中共享变量的值，从而使用共享变量时需要从主内存中重新获取最新的值)

单例懒惰加载时的double-check就是防止指令重排的产物。在下图中变量必须加上volatile修饰，防止指令重排

1. Java的多线程是如何实现的

Java多线程实现方式主要有4种：继承Thread类，实现Runnable接口，实现Callbale接口通过FutureTask包装器来创建Thread线程，使用ExecutorService，callable， Future实现有返回值的多线程



1、RUNNING

(1) 状态说明：线程池处在RUNNING状态时，能够接收新任务，以及对已添加的任务进行处理。   
(02) 状态切换：线程池的初始化状态是RUNNING。换句话说，线程池被一旦被创建，就处于RUNNING状态，并且线程池中的任务数为0！

2、 SHUTDOWN

(1) 状态说明：线程池处在SHUTDOWN状态时，不接收新任务，但能处理已添加的任务。   
(2) 状态切换：调用线程池的shutdown()接口时，线程池由RUNNING -> SHUTDOWN。

3、STOP

(1) 状态说明：线程池处在STOP状态时，不接收新任务，不处理已添加的任务，并且会中断正在处理的任务。   
(2) 状态切换：调用线程池的shutdownNow()接口时，线程池由(RUNNING or SHUTDOWN ) -> STOP

4.TIDYING

(1) 状态说明：当所有的任务已终止，ctl记录的”任务数量”为0，线程池会变为TIDYING状态。当线程池变为TIDYING状态时，会执行钩子函数terminated()。terminated()在ThreadPoolExecutor类中是空的，若用户想在线程池变为TIDYING时，进行相应的处理；可以通过重载terminated()函数来实现。   
(2) 状态切换：当线程池在SHUTDOWN状态下，阻塞队列为空并且线程池中执行的任务也为空时，就会由 SHUTDOWN -> TIDYING。   
当线程池在STOP状态下，线程池中执行的任务为空时，就会由STOP -> TIDYING。

5、 TERMINATED

(1) 状态说明：线程池彻底终止，就变成TERMINATED状态。   
(2) 状态切换：线程池处在TIDYING状态时，执行完terminated()之后，就会由 TIDYING -> TERMINATED。

7、多个线程操作同一个资源会出现什么现象，怎么避免这些问题（线程安全的措施） 线程安全问题？线程安全问题产生的原因是由于jvm底层编译时对代码的优化造成的，包括指令重排和线程对热点数据的缓存，多线程下代码原子性的破坏。可以使用sychronized关键字或者并发包下的Lock对象来对线程进行加锁操作。Sychronized关键字可以保证原子性和可见性，volatile可以保证可见性和指令重排。使用锁加上volatile可以保证可见性，原子性和防止指令重排。但是也不全是通过加锁来解决这个问题。通常的解决方案还包括CAS和volatile关键在来保证线程安全。但是只能在并发量比较低的情况下比较有效，不然循环cas操作不成功会浪费cpu资源。

8、进程和线程有什么区别

进程和线程的**根本区别**是进程是操作系统资源分配的基本单位，而线程是处理器任务调度和执行的基本单位。另外区别还有资源开销、包含关系、内存分配（同一进程的线程共享本进程的地址空间和资源，而进程之间的地址空间和资源是相互独立的）、影响关系（进程比线程健壮，线程之间影响比较大）、执行过程（线程不能独立运行，进程可以独立运行）等

9、**线程之间的调度策略** （）

**二面：**

没录音，凭记忆写的一点，不全

1. springboot的启动类有个注解，@SpringBootApplication如果放到别的类上，能启动成功嘛（瞎猜的不行，面试官说行）

可以启动成功，前提是调用SpringApplication.run方法，在这个方法里会创建一个SpringbootApplication对象并运行run方法(new SpringApplication(primarySources)).run(args);

1. 双亲委派模型、为什么要打破，怎么打破

双亲委派模型通俗的将就是类加载时先交由负载加载，父类加载不到再由子类加载。类加载时是通过loadClass方法加载类的，也就是说其加载逻辑是在这个方法中，双亲委派就是在加载类之前先调用父类的loadClass方法，我们只需要重写ClassLoader类的loadClass方法改变其加载逻辑就可以打破其双亲委派机制。例如tomcat中就打破了双亲委派机制，来保证多版本java类的使用，以及同一版本java类的共享使用

1. 垃圾回收，G1收集器和CMS收集器，CMS为什么要stop the world，怎么解决浮动垃圾问题

Stop the word是因为类的在用户进程运行过程中类的引用关系不是静态不变的，其也在发生变化，我们标记的时候一些垃圾对象被清除不是很大的问题就是消耗一些内存但是如果将正常对象在标记为不可达对象从而被误清除可能会造成程序的崩溃。浮动垃圾只能在下一次垃圾回收来清理，可以预留一部分内存来对应浮动垃圾过多占用内存的情况。

不记得了

手撕代码：

给定一个32位正整数 n，你需要找到最小的32位整数，其与 n 中存在的位数完全相同，并且其值大于n。如果不存在这样的32位整数，则返回-1。

示例 1:

输入: 12 输出: 21 示例 2:

输入: 21 输出: -1

**三面：**

1、问实验室项目，10min

聊一下数据库，主要聊关系型数据库mysql

2、自己展开讲一下innodb引擎

问题：现在都使用固态硬盘，是否存在磁盘寻道的问题，B+树的优势如何体现

3、可重复读在innodb里如何实现的

问题：自己举个实际的例子，加间隙锁的例子，锁哪些范围

4、一段sql语句，怎么建立索引来进行查询优化

select \* from table where col1 > 10 and col2 = 20 order by col3

评论：关于理论的部分答得很好，实战的部分都不是很好

5、redis 的zset，跳表的底层数据结构，实现原理，为什么用这个，怎么查找插入

6、设计一个微信朋友圈功能场景题，需要实现的功能：

用户可以加好友，发动态，浏览首页的动态，如果让我来实现该怎么实现这几个功能，底层数据怎么设计数据库表

7、上面那个场景题，动态里的图片具体怎么存储

手撕代码：

1、M个分段有序数组如何进行快速查找一个整数所在的位置，输出<分段index, 分段内的index>,没有返回<-1, -1>

input： <1, 3, 5, 10> <11, 14, 18, 20>

<25, 30, 30, 40>

find 10 return <0, 3>

find 30 return <2, 2>

2、好友关系进行分组

用户， 好友关系。进行分组。

用户：ABCDEFG

好友：AB，CD， FG，CE， AD

分组：{ABCDE}， {FG}

[https://www.nowcoder.com/discuss/499313?channel=-2&source\_id=subject](https://www.nowcoder.com/discuss/499313?channel=-2&source_id=subject" \t "_blank)

1. 自我介绍

2. 简历项目

3. Java

- Java中实现加锁的方式、sychronized、锁升级过程、底层AQS实现原理

Synchronized关键字和java并发包以及CAS操作（不属于真正的加锁，乐观锁），Synchronized关键字底层有JVM进行优化。比如如果加锁对象是局部对象，可能直接就不加锁。比如在加锁时首先加偏向锁，因为大多数时候是不存在锁竞争的，常常是一个线程多次获得同一个锁，因此如果每次都要竞争锁会增大很多没有必要付出的代价，为了降低获取锁的代价，才引入的偏向锁。

偏向锁的升级

无竞争加偏向锁，竞争少用轻量级锁，竞争多用重量级锁。

（因为阻塞线程需要CPU从用户态转到内核态，代价较大，如果刚刚阻塞不久这个锁就被释放了，那这个代价就有点得不偿失了，因此这个时候就干脆不阻塞这个线程，让它自旋这等待锁释放。）

轻量级锁什么时候升级为重量级锁？

如果自旋的时间太长也不行，因为自旋是要消耗CPU的，因此自旋的次数是有限制的，比如10次或者100次，如果自旋次数到了线程1还没有释放锁，或者线程1还在执行，线程2还在自旋等待，这时又有一个线程3过来竞争这个锁对象，那么这个时候轻量级锁就会膨胀为重量级锁。重量级锁把除了拥有锁的线程都阻塞，防止CPU空转

ObjectMonitor() {

\_count = 0; //用来记录该对象被线程获取锁的次数

\_waiters = 0;

\_recursions = 0; //锁的重入次数

\_owner = NULL; //指向持有ObjectMonitor对象的线程

\_WaitSet = NULL; //处于wait状态的线程，会被加入到\_WaitSet

\_WaitSetLock = 0 ;

\_EntryList = NULL ; //处于等待锁block状态的线程，会被加入到该列表

}

AQS 底层原理：

首先关键属性：state标志位，表示获得锁的次数，ExclusiveThread线程表示当前持有锁的线程。有一个fifo双向队列。这三个属性比较重要。跟ObjectMonitor相似。在请求锁的时候会对state进行Cas操作设置值，设置成功代表抢占锁成功，并将ExclusiveThread设置为该线程。这里会有重入，重入失败会进入同步队列中。

- 平时如何实现线程安全，ConcurrentHashMap

注意叙述一下

- 线程状态、线程池有哪些参数、运行机制、四种拒绝策略

就绪，运行，阻塞，等待，死亡。线程池参数：核心线程数量，最大线程数量，空闲保活时间，时间单位，工作队列（阻塞队列），线程工厂，拒绝策略：

AbortPolicy（默认）

该策略下，直接丢弃任务，并抛出RejectedExecutionException异常

DiscardPolicy

该策略下，直接丢弃任务，什么都不做

DiscardOldestPolicy

该策略下，抛弃进入队列最早的那个任务，然后尝试把这次拒绝的任务放入队列

CallerRunsPolicy

该策略下，在调用者线程中直接执行被拒绝任务的run方法，除非线程池已经shutdown，则直接抛弃任务

4. 计网

- 四次挥手过程、Close wait作用、第2、3次挥手能不能合并

https://mp.weixin.qq.com/s?\_\_biz=MzAwNDA2OTM1Ng==&mid=2453143291&idx=1&sn=d22c5bc68b8503998511c317874f4eac&chksm=8cf2dc78bb85556ef7e3e461b6d6a0d1422a312f3e84c184457eab6df07b014451bbef49b092&mpshare=1&scene=1&srcid=&sharer\_sharetime=1591750875536&sharer\_shareid=605b52fecbd4f990f7fcc587be229eee&key=1e758dfedc4d5b90255f96c400713656cd1de223ed5945b904f636b7cefbbc2282c2c66d2ac3bbeaa3b9c235b33c1e5258335428bfcb6ffa37bfc500632228839725a9bebbfc0eeb6dacc0c724fe4de3&ascene=1&uin=Mjk2NzIzNzgyNA%3D%3D&devicetype=Windows+10+x64&version=62090070&lang=zh\_CN&exportkey=Ay%2Fft9ezHxbdgPeqoXmGx3E%3D&pass\_ticket=7O6g4E3e7rUiRrKFxMRayIIoCG2YyVwLvNrtZoeuGhUcMdtoWZJDqJsF6x18fHu%2F

在服务端没有数据要传输的时候2,3次挥手可以合并。

Time waiting是保证服务端发送的fin连接客户端可以收到，并发送ACK。其规定为报文最大存活时间，这个值太小会导致下次tcp连接可能被破坏，太长占用cpu资源没必要。

- HTTP GET和POST区别

1 一般来说GET是获取数据，POST是提交数据的本质上来说GET和POST请求是没有区别 的，都是TCP链接

1. GET传输数据的时候是在URL地址中的、对所有人都是是可见的、是不安全的、是有浏览器缓存记录的.GET只能传输ASCLL字符，不能进行编码
2. HTTP协议并没有对GET和POST的长度做限制，其实是浏览器限制了他们传输大小.因为浏览器限制了,HTTP协议并没有对GET和POST的长度做限制，其实是浏览器限制了他们传输大小;理论上来说POST的长度是没有限制的，但是受服务器的配置限制或者内存大小的限制;
3. 因为GET是获取数据，所以GET请求是安全且幂等的，是无害的

数据库

- 数据库索引结构、聚簇索引和非聚簇索引区别、B+树结构、为什么不用B树或二叉树

索引的数据结构和具体存储引擎的实现有关，在MySQL中使用较多的索引有Hash索引，B+树索引等，而我们经常使用的InnoDB存储引擎的默认索引实现为：B+树索引

B+tree性质：

1.）n棵子tree的节点包含n个关键字，不用来保存数据而是保存数据的索引。

2.）所有的叶子结点中包含了全部关键字的信息，及指向含这些关键字记录的指针，且**叶子结点本身依关键字的大小自小而大顺序链接**。

3.）**所有的非终端结点可以看成是索引部分**，**结点中仅含其子树中的最大（或最小）关键字**。

4.）**B+ 树中，数据对象的插入和删除仅在叶节点上进行**。

5.）**B+树有2个头指针，一个是树的根节点，一个是最小关键码的叶节点**。

聚簇索引：将数据存储与索引放到了一块，找到索引也就找到了数据

B树只适合随机检索，而B+树同时支持随机检索和顺序检索；

B+树空间利用率更高，可减少I/O次数，磁盘读写代价更低。一般来说，索引本身也很大，不可能全部存储在内存中，因此索引往往以索引文件的形式存储的磁盘上。这样的话，索引查找过程中就要产生磁盘I/O消耗。**B+树的内部结点并没有指向关键字具体信息的指针，只是作为索引使用，其内部结点比B树小**，**盘块能容纳的结点中关键字数量更多，一次性读入内存中可以查找的关键字也就越多，相对的，IO读写次数也就降低了**。**而IO读写次数是影响索引检索效率的最大因素；**

B+树的查询效率更加稳定。B树搜索有可能会在非叶子结点结束，越靠近根节点的记录查找时间越短，只要找到关键字即可确定记录的存在，其性能等价于在关键字全集内做一次二分查找。而在B+树中，顺序检索比较明显，随机检索时，任何关键字的查找都必须走一条从根节点到叶节点的路，所有关键字的查找路径长度相同，导致每一个关键字的查询效率相当。

B-树在提高了磁盘IO性能的同时并没有解决元素遍历的效率低下的问题。**B+树的叶子节点使用指针顺序连接在一起，只要遍历叶子节点就可以实现整棵树的遍历**。而且**在数据库中基于范围的查询是非常频繁的，而B树不支持这样的操作**。

增删文件（节点）时，效率更高。因为B+树的叶子节点包含所有关键字，并以有序的链表结构存储，这样可很好提高增删效率。

B+树在满足聚簇索引和覆盖索引的时候不需要回表查询数据。

在B+树的索引中，叶子节点可能存储了当前的key值，也可能存储了当前的key值以及整行的数据，这就是聚簇索引和非聚簇索引；在InnoDB中，只有主键索引是聚簇索引

- 事务四大特性、事务隔离级别以及分别解决什么问题

一致性，原子性，隔离性，持久性。（读未提交，读已提交，可重复读，串行化）脏读，不可重复读，幻读。

- 幻读、如何解决幻读

幻读的解决，通过next-key lock锁解决幻读问题；一致性读用的是当前读即也就是快照读。

MySQL是如何实现RR和RC的 ？

在innodb的RC或RR模式下，快照读读不到事务未提交的数据。具体是通过***Readview***实现。术语是readview，用白话说就是，**记录了未提交事务的表**，事务会依据该表选择合适的行版本号进行快照读操作，具体选择版本号的方法是**挑选除readview表的版本号外的*最新版本号***。但需要注意的是，RC级别下readview表会随着其他未提交事务的提交而发生变动，但RR不会。当隔离级别为RC，事务一开始便生成readview表，而若是事务执行期间其他事务发生提交操作，此时readview也会相应地改变，之后的快照读会依照改变后的readview表读取最新提交版本的数据，所以RC下会有不可重复读问题。**当隔离级别为RR，事务执行一条语句时才生成readview**，**且之后不会再变动，就克服了不可重复读的问题。（不是读取最新的版本号，而是之前生成的快照中的数据）**

行锁和间隙锁

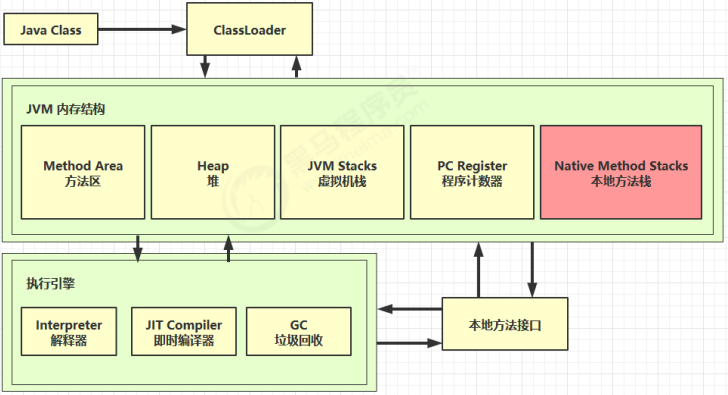
1. 手撕算法（https://leetcode-cn.com/problems/search-rotate-array-lcci/）

https://www.nowcoder.com/discuss/486577?type=all&order=time&pos=&page=1&channel=-2&source\_id=search\_all

* 自我介绍，谈笑风生。
* 项目聊了10分钟。
* 对多态的理解。
* 多线程怎么搞？Thread，Runnable，线程池。
* HTTP Get与Post区别。
* HTTPS聊一下。

HTTPS为了兼顾安全与效率，同时使用了对称加密和非对称加密。数据是被对称加密传输的，对称加密过程需要客户端的一个密钥，为了确保能把该密钥安全传输到服务器端，采用非对称加密对该密钥进行加密传输，总的来说，对数据进行对称加密，对称加密所要使用的密钥通过非对称加密传输。

一个HTTPS请求实际上包含了两次HTTP传输（服务端保存一对公匙、私匙对）：第一次客户端发送https请求，服务端会将公匙发送给客户端。客户端会随机生成一个数作为对称加密的公匙，并用服务端发送的公匙进行加密发送给服务端，服务端接收到之后用自己的私匙进行解密拿到客户端发送的对称加密的公匙。之后的传输就按照对称加密算法来机型加密传输。

* 对JVM的理解。这里谈到了编译的问题。
* 
* 聊一聊JVM的编译怎么搞的？
* 聊聊熟悉的垃圾收集器。
* MySQL事务，隔离级别，隔离级别的实现方式。
* 进程线程区别。
* 代码：求一个数的平方根，精确到0.01。

面试官太棒了，写代码一直指引，后来慢慢写到了最后。。但是不知道为啥没测试成功。面试官说大概是这样的，啊，太菜了。。希望有下一面啊。。

233333，如果还有面试，继续更吧。。

------------------------------------------------------------------------------------------------------------------------------------------------------------

啊。。[猿辅导](/jump/super-jump/word?word=猿辅导" \t "_blank)面试体验太好了。面试官贼棒。

没有自我介绍，后面的问题全是根据项目来问的，包含 Redis 和 Kafka 的东西，Kafka 聊了很久。

问了一下 MySQL 的幻读以及 For UPDATE 会不会触发间隙锁，这里没答起。

问了一下 Spring中的问题，一开始也尬住了，面试官提示想想 AOP，然后回答了。

代码：LeetCode143。写完之后快没时间了，面试官就问了一下我如何去测试这个代码。

[https://www.nowcoder.com/discuss/470045?type=all&order=time&pos=&page=1&channel=-2&source\_id=search\_all](https://www.nowcoder.com/discuss/470045?type=all&order=time&pos=&page=1&channel=-2&source_id=search_all" \t "_blank)

[猿辅导](/jump/super-jump/word?word=猿辅导" \t "_blank)流程这次感觉效率很快，8月17号笔试，26号一面，9月2号二面，4号三面，昨天晚上(10号)意向书，当时首先是HR小姐姐加了微信，我当时感觉是不是凉了要捞我啊 ，然后去邮件箱里刷新了一下邮件，哇，意向书，幸福来的太突然了。有没有了解过[猿辅导](/jump/super-jump/word?word=猿辅导" \t "_blank)的产品啊之类的话题。

总之，十分感谢[猿辅导](/jump/super-jump/word?word=猿辅导" \t "_blank)给了offer，下面是三面的面经，不太详细，祝大家早日拿到心仪的offer!

* **一面**
* 线程池用完之后还没有释放，核心线程与非核心线程的状态
* 各缓存中间件的区别

Mencache不支持持久化

* Redis的缓冲穿透、缓存击穿
  + 缓存击穿，高并发访问，热点key
* JVM中所有OOM的情况
  + 堆溢出、方法区、栈溢出、本地方法栈溢出

栈，堆，方法区

* 异常与错误

Error和exeption exeption又分为runtimeException和除runtimeException之外的异常

Error一般是jvm出现重大错误，例如栈溢出，堆溢出，类找不到

RuntimeException是在运行时由于代码逻辑问题而出现的异常

* + 运行时异常：NullPointerException
  + 非运行时异常：可预知的，编译器负责检查，IOException
* 算法题  
  1. 九宫格数字转字母组合  
  2. 两个有序链表合并
* **二面**
* 算法题
* 1. 十进制转十六进制  
  2. 二叉搜索树转双向链表
* **三面**
* 不让说自己的秒杀项目，因为听得太多了，其他项目二选一
* java NIO
* 进程通信方式

管道（父子进程之间进行通信），FIFO（无关进程之间通信），消息队列（消息的链接表），信号量（用于进程同步和互斥），共享内存（指两个或多个进程共享一个内存区域，比较快的一种方式），套接字（其可以网络中不同主机之间相互通信）；

* Redis为什么快，为什么不用多线程
* 跳表结构
* 一个跳表，应该具有以下特征：

一个跳表，应该具有以下特征：

1. 一个跳表应该有几个层（level）组成；
2. 跳表的第一层包含所有的元素；
3. 每一层都是一个有序的链表；
4. 如果元素x出现在第i层，则所有比i小的层都包含x；
5. 第i层的元素通过一个down指针指向下一层拥有相同值的元素；
6. 在每一层中，-1和1两个元素都出现(分别表示INT\_MIN和INT\_MAX)；
7. Top指针指向最高层的第一个元素

**给定一个有序的链表。**

**2、选择连表中最大和最小的元素，然后从其他元素中按照一定算法（随机）随即选出一些元素，将这些元素组成有序链表。这个新的链表称为一层，原链表称为其下一层。  
3、为刚选出的每个元素添加一个指针域，这个指针指向下一层中值同自己相等的元素。Top指针指向该层首元素  
4、重复2、3步，直到不再能选择出除最大最小元素以外的元素。**

<https://www.cnblogs.com/likui360/p/6124640.html>

* 算法题  
  1. 数据库分表后，给定某一字段查找范围，如何快速定位查哪些表？  
  2. 好友关系，并查集合并[https://www.nowcoder.com/discuss/506809?type=all&order=time&pos=&page=1&channel=-2&source\_id=search\_all](https://www.nowcoder.com/discuss/506809?type=all&order=time&pos=&page=1&channel=-2&source_id=search_all" \t "_blank)

作者：我永远喜欢平泽唯  
链接：[https://www.nowcoder.com/discuss/487920?channel=1009&source\_id=subject](https://www.nowcoder.com/discuss/487920?channel=1009&source_id=subject" \t "_blank)  
来源：牛客网

一面8月8号 45min

特别基础的问题，TCP握手，输入url的过程等等

1.还是先从url拿到目标ip对应的域名，通过各级域名服务器查找域名所对应的ip地址，拿到ip地址之后根据地址解析协议拿到ip地址对应的mac地址以进行数据包的转发，这里的数据包是经过层层包装的数据包，首先经过tcp/ip层,mac层的包装进行。这时候需要经过交换机根据mac地址决定从哪个端口发出，然后就进入到网络中。之后经过路由器的ip寻址逐步发送到相应的目标服务器，服务器经过层层拆包去掉ip头部这些信息最终拿到http数据段，解析资源路径拿到所需的资源。如果这里用的是tcp协议还需要进行三次握手进行连接，

[算法题](/jump/super-jump/word?word=算法题" \t "_blank)：括号匹配（[leetcode](/jump/super-jump/word?word=leetcode" \t "_blank)22）

对脚线遍历（[leetcode](/jump/super-jump/word?word=leetcode" \t "_blank)498）

二面：8月15号 51min

1.自我介绍

2.项目（秒杀和一个消息队列的轮子）

3.面试官对消息队列很感兴趣，聊得很细

4.如果有几百万消息堆积，如何处理

5.如何保证消息消费的可靠性，顺序性

6.rabbitmq和kafka消息队列的模型

7.做题：[leetcode](/jump/super-jump/word?word=leetcode" \t "_blank)306 累加数（写了好久，总有细节出问题）

8.问**netty**，说一些netty的消息处理流程

9.netty为什么高效，以及netty多线程模型（关于netty基本都是自己随意发挥，不过面试官反映不错）

10.好像还有啥来着忘了

二面是三次面试中感觉最好的一次

三面：8月23号  40min

1.自我介绍（我的自我介绍一般不会超过一分钟）

2.面试官让选一个自己觉得最好的项目来讲（ 我那个垃圾项目，最后就讲讲秒杀吧，我这秒杀连分布式都没有）

3.扣细节，最难受的一个地方就是问我项目中一致性问题是怎么解决的。我有点把自己项目搞晕了，没回答好。但是最后也还是给出了个答案。

4.写题，和我字节三面写的第一个[算法题](/jump/super-jump/word?word=算法题" \t "_blank)一模一样，但是我居然没有把我字节三面的那个题搞懂，幸好之后写出来了。然后面试官要求我不断优化。然后就结束了。

找[二叉树](/jump/super-jump/word?word=二叉树" \t "_blank)中序遍历的后继节点。

[https://www.nowcoder.com/discuss/492623?type=post&order=time&pos=&page=0&channel=1009&source\_id=search\_post](https://www.nowcoder.com/discuss/492623?type=post&order=time&pos=&page=0&channel=1009&source_id=search_post" \t "_blank)

[猿辅导](/jump/super-jump/word?word=猿辅导" \t "_blank)面试体验特别好，三个面试官都很nice

一、[猿辅导](/jump/super-jump/word?word=猿辅导" \t "_blank)一面 08/08 50min

自我介绍   
项目   
公钥和私钥分别解决什么问题？（公钥解决证书问题，私钥不清楚）   
进程和线程

虚拟存储器，指具有调入和置换功能的，从逻辑上对内存加以扩充的一种存储器系统。主要内存不足，以及进程运行时，程序和数据不需要全部调入内存，按需分批调入和换出。这些暂时存放在程序的外存和内存决定了虚拟存储器的逻辑容量。用户看到的内存远远大于实际的内存。当出现访问的页和段没有在内存中时发生缺页中断或者缺段中断。地址变化机构产生和处理中断。访问快表，访问内存；都不在，中断，保存cpu现场，从外存中找缺页调入内存，在内存满的情况下，选择一页换出腾出内存。涉及到分配的最小物理块数，何时调入（预调入，请求时调入）是否需要写回磁盘（修改写回，不修改不写回）置换算法（最近最久未使用，最少使用，Clock算法）

抖动，进程太多，分配给每个进程的物理块太少，致使出现频繁的缺页。（减少程序的运行，工作集算法，局部置换：只从本程序中的内存中置换）  
分页机制   
JVM内存机制

**程序的装入**：绝对定位装入（单程序），重定位装入 （多程序，也是物理地址等于初始物理地址+逻辑地址），动态装入（地址转换到程序真正执行时才进行，换入换出导致前后内存地址不同，需要重定位寄存器的支持）

**连续分配：内存的分区：固定分区**（分成大小相等或者不等的不变的分区，需要一张分区分配表来记录分区起始、终止地址和状态）分区大于所要分配程序的大小造成内存浪费；

**动态分区**，分区数据结构：空闲分区表，空闲分区链；——**动态分区算法**：顺序搜索，索引搜索（需要将空闲分区按照大小分类，相同大小的建立一张空闲链表）小碎片不能紧凑的问题--**动态可重定位分区算法**：在动态分区算法中加入了紧凑功能（就是将分散的小分区合并成大分区），因为有重定位寄存器的支持，所以不用担心内存起始变化带来的影响，其会帮忙完成程序的新起始地址的修改。

对换：原因：内存空间有限导致，某些进程阻塞，而另一些进程内存不足需要程序在内存中的换入换出。整体换出，部分换出

**离散分配：原因：紧凑功能会付出很大的开销，如果将程序放入不邻接的内存中，就无须进行紧凑从而也可充分的利用内存空间。**

**分页存储，分段存储，段页式存储：**

**分页：进程分页，内存分块，最后一页填补满一个内存块形成碎片称为页内碎片，页面大小应该适中，页内碎片和页表（实现页号到物理块号的映射，由于比较频繁，直接采用硬件地址变换机构来实现）过长造成的空间浪费。页表项数过大造成占用内存过大，可以采用多级页表的方式来处理。**

**分段存储：程序中的程序都可分为若干段，比如主程序，子程序，这些有逻辑关系，包括信息保护（只读，读写）。程序分段，跟分页一样维持段表，映射逻辑地址和物理地址。页是物理单位，段是逻辑单位，按大小分和按逻辑分。页大小固定由系统决定，段大小不固定，由用户决定。**

**段页式：先把程序分段再把程序分页，（其地址结构由段号，段内页号，页内地址）段表指向页表，页表指向物理地址。一般都有高速缓存寄存器，或通过页号，段号以及段号加页号来完成。**

分区的内存分配和回收  
HTTP/HTTPS

算法：单链表奇偶拆分

二、 [猿辅导](/jump/super-jump/word?word=猿辅导" \t "_blank)二面 08/13 48min

二面面试官上来先和我聊会学业问题，再开始面试，体验很好

自我介绍   
项目   
线程池原理（提交任务过程，拒绝策略，shutdown()和shutdownNow() 的区别）   
MySQL索引的原理   
联合索引（a,b,c) where a = 100 and b > 2 and c > 3 能使用哪些索引   
4个隔离级别，  
算法：实现 topK 并分析时间复杂度   
  
[猿辅导](/jump/super-jump/word?word=猿辅导" \t "_blank)三面 08/23 50min   
自我介绍   
项目   
String内部有哪些属性   
String 的concat()内部实现   
ArrayList实现，缩容问题   
进程线程协程，**协程**切换时寄存器上下文和栈保存在哪   
HTTPS原理   
没录音，其他忘了

算法：洗牌算法变种

[https://www.nowcoder.com/discuss/470051?type=post&order=time&pos=&page=1&channel=1009&source\_id=search\_post](https://www.nowcoder.com/discuss/470051?type=post&order=time&pos=&page=1&channel=1009&source_id=search_post" \t "_blank)  
来源：牛客网

1. 自我介绍
2. 项目介绍
3. 操作系统：
   1. 进程间通信方式，以及对应的应用场景

不同机器用socket,进程同步与互斥，共享内存（交换数据量比较大且）

1. 计网：
   1. 网络层的协议有哪些
   2. http 和 https 的区别
2. java 容器：
   1. ArrayList 线程安全吗？为什么？
   2. HashMap 的底层结构，为什么用[红黑树](/jump/super-jump/word?word=红黑树" \t "_blank)
3. JVM：
   1. 为什么要有新生代和老年代
   2. 对象什么时候进入老年代
4. 数据库：
   1. 为什么使用 b+ 树

此时过去了 17 分钟 ……

1. 算法：
   1. [二叉树](/jump/super-jump/word?word=二叉树" \t "_blank)找到最后一层的第二个节点，若不存在则抛出一个异常。（用了 11 分钟）
   2. 单调递增的数组，找到 i∈[m,n)i \in [m,n)i∈[m,n) 的个数。（忘记看时间了。很快做出来了，但漏考虑了一个边界条件）
2. 反问

# 二面 8.14 (50分钟)

1. 自我介绍
2. 实习过吗
3. java：
   1. 双亲委派机制 (抱歉)
4. JVM：
   1. 内存模型
   2. 类加载机制 （抱歉）
   3. CMS收集器 （抱歉）
   4. 什么时候会发生 stackoverflow
5. java 容器：
   1. hashmap 为什么 1.8 用尾插法就不会死循环了
   2. 讲讲 arraylist 的实现 (讲了底层结构，扩容机制)
   3. 讲讲 arraylist 的扩容过程
6. 看源码有哪些收获
7. mysql 数据库：
   1. 隔离级别
   2. 可重复读解决了什么问题

此时大约过去了 24 分钟。

[算法题](/jump/super-jump/word?word=算法题" \t "_blank)：给定一个字符串，判断是否是可累加数列。例如："199100199"，由于 1 + 99 = 100, 99 + 100 = 199，所以是。

反问：

* 1. 反馈：个人特点鲜明，对学过的东西掌握比较深，但知识面不够广，面试可能会吃亏
  2. 部门业务： 三面的时候才确定部门
  3. 晋升机制：扁平化管理，职级只跟薪酬挂钩，按绩效来
  4. [职业规划](/jump/super-jump/word?word=职业规划" \t "_blank)：前 3 年提升技术，和软实力（沟通能力，任务规划能力等）。后面根据兴趣选择发展方向

链接：[https://www.nowcoder.com/discuss/470466](https://www.nowcoder.com/discuss/470466" \t "_blank)

· 自我介绍

· **Java基础**

* 创建对象的方式
* @Autowired注解原理

AutowiredAnnotationBeanPostProcessor，InjectionMetadata他的postMergeBeanDefinition方法中会判断该beandefinition包含的class中是否有注解，如果有的话就可以解析器类信息进行注入，并且会以beanname为key进行缓存，以便下次可以直接使用

String,StringBuilder,StringBuffer区别

· **Spring**

* IOC相关
* AOP理解
* cglib与proxy

· **计网**

* 为什么不两次和四次握手
* 挥手不三次
* 在服务端接收到fin请求时，其可能还需要发送数据
* 三次握手能否携带数据包 ：TCP协议建立连接的三次握手过程中的第三次握手允许携带数据，带有SYN标志的过程包不允许携带数据，第三次应该是可以的。

· **数据库**

* 隔离级别
* mysql默认隔离级别
* 可重复读存在的问题 可能会出现幻读的情况。
* 间隙锁的底层实现

在有索引的时候如果查找的记录不唯一，这时候会用间隙锁锁定一个范围。

· **算法**

* 求[二叉树](/jump/super-jump/word?word=二叉树" \t "_blank)第K层节点数
* 已知电影开始与结束时间，看最多场电影

1. 自我介绍   
   2.项目   
   3.给一个[二叉树](https://www.nowcoder.com/jump/super-jump/word?word=%E4%BA%8C%E5%8F%89%E6%A0%91" \t "https://www.nowcoder.com/discuss/_blank)，返回max(深度，宽度)   
   4.线程、进程

上来就两道题

1.第一道是求一棵树的最大距离，也就是某两个节点的最大距离

2.求出一个链表的第一个环

我不到20分钟就做完了，然后还叫我做第三道题：

3.求二叉搜索树的最近公共祖先，做到一半叫我讲思路，然后跳过

作者：黑色颠沛  
链接：[https://www.nowcoder.com/discuss/470387](https://www.nowcoder.com/discuss/470387" \t "_blank)  
来源：牛客网

· 自我介绍

· 解释一下volatile关键字，原子类底层CAS怎么实现的

· HashMap，get过程，为什么之前不用[红黑树](/jump/super-jump/word?word=红黑树" \t "_blank)

· JVM内存，为什么要有S区，新生代有啥特点

· Java怎么判断哪些对象是需要回收的

· sychronized和Lock的区别以及各自优势

· 数据库主键索引非主键索引的区别

· 项目HTTPS的握手过程，能被中间人劫持吗？

· [算法题](/jump/super-jump/word?word=算法题" \t "_blank)：

[LeetCode求两个链表只和，从头到尾相加即可](/jump/super-jump/practice?questionId=1008772" \t "_blank)

· · [算法题](/jump/super-jump/word?word=算法题" \t "_blank)：给定一个数n，输出1-n的字典[排序](/jump/super-jump/word?word=排序" \t "_blank)数组，如n=160，输出[1,10,100,101,....,109,11,110,....,160]

作者：童大仙  
链接：[https://www.nowcoder.com/discuss/470197?](https://www.nowcoder.com/discuss/470197?" \t "_blank)  
来源：牛客网

1、自我介绍

2、xms xmx分别代表什么 初始堆内存和最大堆内存。

3、为什么要有这两个

4、为什么要用分代算法

5、为什么新生代要用复制算法，老年代要用标记清除/标记整理？

6、jvm的各个区域说一下

7、怎么判断对象存活？

8、GC ROOTS具体是怎么遍历的？

9、TCP连接讲一下

10、什么是面向连接的 ：

建立一条虚电路，与非面向连接是想对的概念。即在每次传输数据在连接没有断开之前都不会再创建新的连接。

如果一种服务具有下列特征，就认为它是面向连接的：

1、建立一条[虚电路](https://baike.baidu.com/item/%E8%99%9A%E7%94%B5%E8%B7%AF" \t "https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E8%BF%9E%E6%8E%A5/_blank)（比如[3次握手](https://baike.baidu.com/item/3%E6%AC%A1%E6%8F%A1%E6%89%8B" \t "https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E8%BF%9E%E6%8E%A5/_blank)）

2、使用排序

3、使用确认

4、使用[流量控制](https://baike.baidu.com/item/%E6%B5%81%E9%87%8F%E6%8E%A7%E5%88%B6/3441910" \t "https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E8%BF%9E%E6%8E%A5/_blank)。流量控制的类型有：缓冲、窗口机制和拥堵避免

滑动窗口：A向B发送数据。在连接建立时，B告诉了A：“**我的接收窗口是rwnd = 400**”（这里的rwnd表示receiver window）。因此，发送方的发送窗口不能超过接收方给出的接收窗口的数值。请注意：TCP的窗口单位是字节，不是报文段； 如果发送方把数据发送得过快，接收方可能会来不及接收，这就会造成数据的丢失。所谓流量控制就是让发送方的发送速率不要太快，要让接收方来得及接收；

11、怎么保证传输的可靠性 校验和，序列号和确认应答，超时重传，滑动窗口流量控制，拥塞控制。

12、我传12345678个报文段过去，从第五个开始丢失了，发送方怎么知道？超时重传，和快速重传

13、滑动窗口、拥塞窗口讲一下 发送方去拥塞窗口和滑动窗口的最小值来发送数据，拥塞窗口是发送方使用的流量控制（网络拥塞），滑动窗口是接受方使用的流量控制（发送方和接受方速度不匹配）

14、拥塞窗口的慢开始、拥塞避免、快重传、快恢复

所谓拥塞控制就是防止过多的数据注入到网络中，这样可以使网络中的路由器或链路不致过载。拥塞控制是一个全局性的过程，和流量控制不同的是，流量控制往往指的是点对点通信量的控制，是个端到端的问题；

发送方维持一个叫做拥塞窗口cwnd（congestion window）的状态变量。拥塞窗口的大小取决于网络的拥塞程度，并且动态地在变化。发送方让自己的发送窗口等于拥塞窗口，另外考虑到接受方的接收能力，发送窗口可能小于拥塞窗口。

慢开始算法的思路就是，不要一开始就发送大量的数据，先探测一下网络的拥塞程度，也就是说由小到大逐渐增加拥塞窗口的大小。

拥塞避免算法使得CWND按照线性方式增加，从而**减缓**其扩大，即不像慢开始那样增加那么快。（由于TCP采用的是**累计确认机制当一些报文段全部到齐之后会发送确认ACK**，即当接收端收到比期望序号大的报文段时，便会重复发送最近一次确认的报文段的确认信号，我们称之为冗余ACK（duplicate ACK）。  
报文段1成功接收并被确认ACK 2，接收端的期待序号为2，当报文段2丢失，报文段3失序到来，与接收端的期望不匹配，接收端重复发送冗余ACK 2。）

1.传输超时

2、接收到重复的确认报文段（在发生拥塞时）

如果发生第一种情况，就重新调整慢启动门限的值ssthresh，故而拥塞控制再次进入慢启动阶段。

如果为第二种情况，则启用快速重传和快速恢复（如果发生在重传定时器溢出后当作第一种情况处理）

如果发生上述第二种情况，拥塞控制算法要怎么判断此时是真的发生了拥塞呢？具体做法：快重传算法要求首先接收方收到一个失序的报文段后就立刻发出重复确认，发送端如果连续收到了3个重复的确认报文段，就认为拥塞发生了。当收到第3个重复的确认报文时，然后立即重传丢失的报文段

当收到3个重复ACK时，把ssthresh设置为cwnd的一半，把cwnd设置为ssthresh的值加3，然后重传丢失的[报文段](https://baike.baidu.com/item/%E6%8A%A5%E6%96%87%E6%AE%B5/2603971" \t "https://baike.baidu.com/item/%E5%BF%AB%E9%87%8D%E4%BC%A0%E5%92%8C%E5%BF%AB%E6%81%A2%E5%A4%8D/_blank)，加3的原因是因为收到3个重复的ACK，表明有3个“老”的数据包离开了网络

再收到重复的ACK时，[拥塞窗口](https://baike.baidu.com/item/%E6%8B%A5%E5%A1%9E%E7%AA%97%E5%8F%A3" \t "https://baike.baidu.com/item/%E5%BF%AB%E9%87%8D%E4%BC%A0%E5%92%8C%E5%BF%AB%E6%81%A2%E5%A4%8D/_blank)增加1。

1. 当收到新的数据包的ACK时，把[cwnd](https://baike.baidu.com/item/cwnd/5905400" \t "https://baike.baidu.com/item/%E5%BF%AB%E9%87%8D%E4%BC%A0%E5%92%8C%E5%BF%AB%E6%81%A2%E5%A4%8D/_blank)设置为第一步中的ssthresh的值。原因是因为该ACK确认了新的数据，说明从重复ACK时的数据都已收到，该恢复过程已经结束，可以回到恢复之前的状态了，也即再次进入拥塞避免状态

进程调度算法：

先来先去服务

　　2.时间片轮转法

　　3.多级反馈队列算法

　　4.最短进程优先

　　5.最短剩余时间优先

　　6.最高响应比优先

1. 7.多级反馈队列调度算法

15、[算法题](/jump/super-jump/word?word=算法题" \t "_blank)1：

[二叉树的非递归中序遍历](/jump/super-jump/practice?questionId=1008937" \t "_blank)

16、[算法题](/jump/super-jump/word?word=算法题" \t "_blank)2：一个链表，给定一个目标值，比目标值大的节点去到链表后面，要求不改变相对顺序

比如1 3 2 1 3 2 1 给定目标值2，返回链表1 2 1 2 1 3 33.

作者：WenKoKu  
链接：[https://www.nowcoder.com/discuss/470199](https://www.nowcoder.com/discuss/470199" \t "_blank)  
来源：牛客网

C++

操作系统

进程、线程

进程通信

锁

线程死锁和进程死锁区别

行锁和表锁如何死锁

虚拟内存

内存置换

数据库

索引

B+树

计算机网络

TCP和UDP区别，解决了什么问题，应用场景是什么

TCP可靠性，滑动窗口，拥塞控制

基于UDP协议应用协议，解决了什么问题，应用哪些场景

ssl协议安全原理，解决了什么问题，应用场景

证书解决了什么问题，分为几种证书，分别应用场景

对称加密和非对称加密分别解决什么问题和应用场景

RSA的公钥和私钥分别解决什么问题，应用场景

HTTP协议，解决的问题和场景

请求哪几种方式，分别解决什么问题，哪些应用场景

回应码有哪些，分别表示什么意思

GET和POST的区别，什么场景

HTTP1.0、1.1有什么区别

长连接和短连接区别，解决了什么问题，应用场景

长连接：多用于操作频繁，点对点的通讯，而且连接数不能太多的情况。每个TCP连接的建立都需要三次握手，每个TCP连接的断开要四次握手。

web网站的http服务一般都用短连接。因为长连接对于服务器来说要耗费一定的资源。像web网站这么频繁的成千上万甚至上亿客户端的连接用短连接更省一些资源。

长连接的timeout有什么用，解决了什么问题，有没有什么其他的应用场景

HTTP各请求方式的幂等性，解决什么问题和场景

withdraw的语义是从account\_id对应的账户中扣除amount数额的钱；如果扣除成功则返回true，账户余额减少amount；如果扣除失败则返回false，账户余额不变。

ithdraw请求已经被服务器端正确处理，但服务器端的返回结果由于网络等原因被掉丢了，导致客户端无法得知处理结果

int create\_ticket()

bool idempotent\_withdraw(ticket\_id, account\_id, amount)

create\_ticket的语义是获取一个服务器端生成的唯一的处理号ticket\_id，它将用于标识后续的操作。idempotent\_withdraw和withdraw的区别在于关联了一个ticket\_id，一个ticket\_id表示的操作至多只会被处理一次，每次调用都将返回第一次调用时的处理结果。

作者：爱尾巴  
链接：[https://www.nowcoder.com/discuss/470194](https://www.nowcoder.com/discuss/470194" \t "_blank)  
来源：牛客网

先自我介绍一下吧

介绍一下自己的项目里面有什么亮点或者说让你印象深刻的东西

[Redis](/jump/super-jump/word?word=Redis" \t "_blank)分布式锁是怎么实现的

如果多个线程同时要setnx，redis会怎么做

你用的Java版本是什么，jdk1.8有哪些新特性

Restful了解吗？

get和post有哪些区别

post不用request body可以吗

HTTP状态码知道哪些？304知道是什么吗？（我说成302了QAQ）

HTTP里面数据的格式是什么样的？

HTTP里面可以放哪些数据？

如果我从前端给服务端传文件该怎么办？

跨域了解吗？知道是什么原理吗？

除此之外有一堆网络相关的问题，有些很深入，没答好

================到此为止20分钟左右====================

编程一：多线程打印数字

编程二：

[单链表反转](/jump/super-jump/practice?questionId=23286" \t "_blank)

（先创建了一个临时节点，然后逐个头插），如果不用中间变量怎么做？（指针指向尾结点然后头插）如果不更改链表结构的话该怎么做？（时间不够了，说了一下思路）如果用递归怎么做？（一时间没想起来，说的乱七八糟的）

作者：许愿二面  
链接：[https://www.nowcoder.com/discuss/470319](https://www.nowcoder.com/discuss/470319" \t "_blank)  
来源：牛客网

一面8月7号，总计1个小时

整体流程：自我介绍、项目、基础、两道[算法题](/jump/super-jump/word?word=算法题" \t "_blank)

（1）自我介绍

（2）挑一个项目讲

（3）http流程

（4）epoll

Bio: 线程监听socket对象，当数据达到时，会唤醒监听该socket对象的线程，并进行数据的读写。

nio：上述很明显是一个线程只能监听一个socket对象，selector的解决方案是使用seletor对象存放所有的socket对象。当有一个socket对象接受到数据后就会唤醒持有selector对象的线程，但是其不知道是哪个socket接受了数据。需要遍历所有socket。

Epoll在此seletor的基础上维持了一个就绪socket对象列表，在就绪列表中的所有socket对象就是已经接收到数据的对象。

（5）reactor

（6）数据库事务，隔离级别（包括什么是脏读、幻读和可重复读）

（7） 两道关于树的题，一道二叉搜索树转双向链表、一道二叉搜索树给定节点求中序遍历的下一个节点（这个题有点自己绕进去了，面试官提醒才做出来）

二面8月14号，下午5点~6点

（1） 自我介绍

（2）将项目

（4）项目中的编码解码如何做的

（5）进程间通信有哪些

（6）共享内存实现

（7）**匿名管道和有名管道区别**

（8）**如何创建一个进程（fork）**

（9）父进程提前退出，子进程状态（孤儿进程）

（10）Tcp流程

（11）路由协议有哪几个

（12）如何杀死进程（kill）

（13）Kiil -9和-15 分别是什么（这个记不大清了，分别是***SIGKILL***和***SIGTERM***）

（14）编程题：

（1）求树的最大宽度（这个简单的层序遍历）

（2）复原ip（这个做的有点慢了，最后就写了个大框架）

差不多就这些，都是围绕项目展开的，面试官很好，全程微笑，给人感觉挺好，希望可以让我通过，许愿三面。

作者：godyunfly  
链接：[https://www.nowcoder.com/discuss/469950](https://www.nowcoder.com/discuss/469950" \t "_blank)  
来源：牛客网

2020.8.7 14:00(40min)

0、自我介绍

1、讲一讲你的项目中的亮点

2、分布式事务如何实现的？（讲一讲底层原理）

3、讲一下MySQL事务的实现原理和ACID特性

4、介绍一下undo log和redo log

前滚和回滚

5、介绍一下ThreadLoacl的实现原理

6、如果 ThreadLoaclMap的Key有null的情况呢？

7、介绍一下Java的强引用和弱引用

8、编程题：判断给定的一颗[二叉树](/jump/super-jump/word?word=二叉树" \t "_blank)是否是二叉搜索树

面试官人很好，第一次写的时候脑子卡住了，用了额外的空间，面试官一点拨我才发现问题

然后详细分析了我两次写的代码的时间和空间复杂度，讲一讲原因

最后许愿一个二面



------------------------------------------------------------------------------

更新一下，8.14面了二面

8.14  10：00（70min）

0、自我介绍

1、讲一个你认为最出彩的项目，根据项目提了几个问题

2、问我为什么使用了Spring Cloud的情况下，又使用Dubbo

3、Zk的ZAB协议

4、进程间的通信方式，其中消息队列的方式没答好

这个我说成了是Rabbit MQ之类的消息中间件，但是实际上应该是操作系统内置的消息队列

5、JVM的内存区域划分，每个部分都存放哪些内容

6、OOM一般在什么情况下发生，反映到JVM内部是怎么样的

7、 ThreadLoacl的实现原理（话说这个一面不是问了吗）

8、介绍一下线程池在项目中具体怎么用的？参数什么的如何设置的，理由是什么

9、线程有哪些状态？分别调用什么方\*\*\*变成相应的状态

10、线程池中的空闲线程处于什么状态？（我回答处于等待状态，面试管没有回答我是否正确，请教一下各位大佬）

11、[算法题](/jump/super-jump/word?word=算法题" \t "_blank)：利用循环链表写一个队列

面试官人很好，一直笑眯眯的，哈哈哈哈

作者：sagii  
链接：[https://www.nowcoder.com/discuss/470177](https://www.nowcoder.com/discuss/470177" \t "_blank)  
来源：牛客网

1、ARP协议，怎么工作的、属于哪一层，为什么属于数据链路层，数据链路层会认识ip地址吗

2、ICMP了解吗？是做什么的？ping的具体过程？

3、DNS解析过程

4、了解TCP协议吗

5、UDP协议为什么能实现广播

6、详细说一下tcp的拥塞控制

7、说一说数据库建表的范式规范，什么是外键约束？第一范式第二范式第三范式

8、说自己的项目

9、zk做服务发现，如何发现这个服务是activate的，服务下线的话，zk怎么观察到服务的变化，dubbo接口的服务调整，服务上下线，zk怎么应对。

10、AOP的原理

11、对java的语言特性

12、java虚拟机是平台相关的吗

13、链表[排序](/jump/super-jump/word?word=排序" \t "_blank)

<https://www.nowcoder.com/discuss/470104>

作者：X时节X  
链接：[https://www.nowcoder.com/discuss/470665](https://www.nowcoder.com/discuss/470665" \t "_blank)  
来源：牛客网

57分钟，两道很简单的手写算法，第一次面试写代码有点慌，面试官人挺好，那么尬的场景都憋得住

基本问题：

1.数据库的读写一致性

2.http2.0与之前的区别

3. https的流程，对称加密和非对称加密

4. 循环依赖

5. 死锁的条件和解决方法

6.session与cookie应用的区别

代码题：

1.

[找出只出现一次的数字](/jump/super-jump/practice?questionId=23275" \t "_blank)

（若干方法实现）

2. [二叉树](/jump/super-jump/word?word=二叉树" \t "_blank)最大深度（若干方法实现）

因为树节点输入不知道怎么搞，卡了很久。。。。最后勉强都按要求写了，基本已凉

作者：沛沛#  
链接：[https://www.nowcoder.com/discuss/470666](https://www.nowcoder.com/discuss/470666" \t "_blank)  
来源：牛客网

面试体验很棒，面试官很和蔼

首先面试官自我介绍，然后本菜鸡自我介绍

针对建立提问，简历上的项目，用了什么技术，对于springMVC、MyBatis的理解

[Redis](/jump/super-jump/word?word=Redis" \t "_blank)的持久化，项目中哪些地方用了redis

java内存模型

线程和进程：是否了解协程，本菜鸡不会

代码题：给一个字符串数组，把由相同字母组成的字符串分入同组

ps：代码问题我理解错了，写了个字符串的所有乱序排放，测试的时候发现自己理解错了。非常尴尬，然后面试官就顺着我写的程序进行了提问

作者：人菜就要多学习  
链接：[https://www.nowcoder.com/discuss/470712](https://www.nowcoder.com/discuss/470712" \t "_blank)  
来源：牛客网

[猿辅导](/jump/super-jump/word?word=猿辅导" \t "_blank)服务端开发一面45min

全程扣着计网不放，我也不是学前端的，咋一个劲怼计网

1.自我介绍

2.介绍一下项目

3.用到md5加密了，过程说一下？

4.md5只用一次不行吗？

5.md5可逆不？

6.除了md5还了解别的加密技术吗？

7.说说对称加密和非对称加密？

8.说说https加密过程

9.说一下验证证书合法性的过程(信任链)

10.一个证书只能加密一个域名吗？子域名可以不？(可以)

11.https是对请求头还是请求体加密？

12.除了get和post，还用到别的方法了吗？

13.put是干嘛的？它有请求体吗？

14.请求头请求体都包括啥？

15.content-length一定有吗？如果数据特别长，一次都能读取吗？那length表示的是每次读取的长度还是总长度？

16.请求体的数据结构？(json,xml之类的)

17.长连接是干啥用的？举个场景说

18.长连接的连接数有限制吗？

19.跨域了解吗？

20.cookie支持跨域吗？

21.cookie可以包括哪些类型的数据？

22.rabbitmq有几种工作模式？

23.自己用的哪种？

24.mq里的订单如何删掉？(设置过期时间)

25.[算法题](/jump/super-jump/word?word=算法题" \t "_blank)

[反转链表，迭代+递归两种都写](/jump/super-jump/practice?questionId=23286" \t "_blank)

作者：张志恒  
链接：[https://www.nowcoder.com/discuss/470707](https://www.nowcoder.com/discuss/470707" \t "_blank)  
来源：牛客网

1 项目 消息加入异步队列超时或者失败，你是怎么处理的

2 redis做持久化了吗，怎么保证rdb的完整性

3 redis淘汰策略

4 redis怎么实现事务

5 数据库 事务隔离级别

6  索引为什么用b+树  不用hash 不用平衡[二叉树](/jump/super-jump/word?word=二叉树" \t "_blank)

7 自己用过的索引讲讲

8 linux top命令 具体CPU负载是怎么计算出来的  （不会）

怎么查看端口 netstate

9 http状态码

10  cookie session讲一下

11 [算法题](/jump/super-jump/word?word=算法题" \t "_blank)

1

[矩阵顺时针翻转90度](/jump/super-jump/practice?questionId=25283" \t "_blank)

2

[从n到m翻转链表](/jump/super-jump/practice?questionId=654" \t "_blank)

作者：涅槃15  
链接：[https://www.nowcoder.com/discuss/470843](https://www.nowcoder.com/discuss/470843" \t "_blank)  
来源：牛客网

## 操作系统

* 进程和线程的区别
* 进程通信方式
* 死锁以及产生条件
* 虚拟内存这一块，针对性问了一些问题，局部性原理等。

## 计算机网络

* TCP、UDP
* UDP怎么实现可靠传输
* https连接过程、http2.0

## java

* hashmap、hashtable区别

## 场景题

* 设计一个客服端登录注册的过程（重点聊了下单点登录设计和cookie,seesion）
* cookie里面保存了什么数据。
* 用户禁用cookie之后怎么保持有状态连接。  
  等等，记不清了

## 算法

1：求[二叉树](/jump/super-jump/word?word=二叉树" \t "_blank)最大高度（非递归的方式）  
2：[1,1,2,2,3,3,4,5,5,6,6,7,7......]有序数组中唯一的数，输出4。要求时间复杂度o(lgn)

作者：牛客641740495号  
链接：[https://www.nowcoder.com/discuss/470743](https://www.nowcoder.com/discuss/470743" \t "_blank)  
来源：牛客网

面试时间54min

1.自我介绍

2.项目介绍

3.mongodb优势，集群搭建，引擎

4.tcp与udp，udp怎么保证可靠传输

5.mysql 事务，隔离级别

1-5时长18min

写题：（我觉得面试官很可以！给了许多提示，都能get到，但是我太菜了）

1.拆分奇偶链表

2.[二叉树](/jump/super-jump/word?word=二叉树" \t "_blank)剪枝

作者：救救孩子吧996  
链接：[https://www.nowcoder.com/discuss/470720?](https://www.nowcoder.com/discuss/470720?" \t "_blank)  
来源：牛客网

本人投的服务端开发，主要语言java

上来先自我介绍。

选择一个做的最好的项目详细说一下，遇到的最大的困难？，怎么解决的？

然后就是直接手撕[算法题](/jump/super-jump/word?word=算法题" \t "_blank)。

第一题

[最长回文子串LeetCode5](/jump/super-jump/practice?questionId=25269" \t "_blank)

相关知识点： [字符串动态规划](javascript: void(0))

相关知识点： [字符串动态规划](javascript: void(0))

先是说思路，然后交流看可不可行。

写了递推公式，面试官提醒应该加条件。最后写出来了，调试的时候一直出错，怕给面试官不好的印象，

就不敢继续调了，因为有个案例老是不通过，问他能不能帮我看下，后来发现是顺序写的有问题，应该从右下角往左上角推。

然后结果对了。

第二题 根据中序遍历和后序遍历还原[二叉树](/jump/super-jump/word?word=二叉树" \t "_blank)

说了思路，递归构造，然后开始写，最后没写出来，不知道具体裁剪边界怎么确定，后来时间快到了，就没写了，

面试官说大致明白我的思路了。

**感谢面试官这么有耐心**，希望能过，许愿二面。

作者：Mao-YK  
链接：[https://www.nowcoder.com/discuss/470787](https://www.nowcoder.com/discuss/470787" \t "_blank)  
来源：牛客网

#### ，自我介绍

#### 2，计算机网络

HTTP和HTTPS的区别，

HTTPS深入讲

TCP和UDP

TCP如何保证可靠性连接

TCP的三次握手和四次挥手，为什么要三次挥手？Time\_wait发生在哪里？为什么设置要两个报文的最大生存时间？

#### 3，[算法题](/jump/super-jump/word?word=算法题" \t "_blank)

找二叉查找树的两个节点相差的最小值

（先序遍历递归查找，维护一个最小值）

手残，算法没写好，把思路跟面试官说了，时间到了

#### 4，反问

作者：chips\_xx  
链接：[https://www.nowcoder.com/discuss/470970](https://www.nowcoder.com/discuss/470970" \t "_blank)  
来源：牛客网

1.项目相关

2.你知道哪些线程池？

3.线程池参数以及线程池的流程

4.线程池的抛弃策略

5.停止线程池，停止时会接新的任务嘛？

6.get和post的区别，post一定会发两个包嘛？

7.还有什么请求方法

8.http与https的区别

9.主键索引和唯一索引的区别

10.索引结构，b+树是怎样的结构，b树与b+树的区别

11.数据库的隔离级别，每种级别会存在什么问题

12.讲一讲gap lock和next-key lock，具体是怎么锁的，比如age=20,锁的是id还是age

13.联合索引最左匹配的使用

14.MVCC

15.redis的数据类型

16.hash,set,zset的使用场景

17.zset内部实现

18.TCP的拥塞控制

19[算法题](/jump/super-jump/word?word=算法题" \t "_blank):

[topK](/jump/super-jump/practice?questionId=44581" \t "_blank)

1. 反问

作者：随心的小新  
链接：[https://www.nowcoder.com/discuss/471164](https://www.nowcoder.com/discuss/471164" \t "_blank)  
来源：牛客网

首先上来万年不变的自我介绍。接着介绍一下项目

1、http的幂等性问题，网络框架以及各层之间的协议都有什么？

2、redis的分布式锁，超时时间以及时间续约？

3、线程安全问题？线程安全问题怎么去解决

4、死锁？死锁的必要条件？怎么去预防续约以及死锁之后的解决方法？

5、synchronized之后的1.6之后的升级过程，以及与lock的区别？

6、RbbbitMQ的一个重复推送问题怎么解决？它都有哪种模式？

大概就是这么些东西吧。

最后放个大招。手撕八皇后问题。有点猛

作者：清凰  
链接：[https://www.nowcoder.com/discuss/471105](https://www.nowcoder.com/discuss/471105" \t "_blank)  
来源：牛客网

1.自我介绍

2.项目的RBAC权限模型，提了几个场景题

3.SpringCloud组件，以及一些问题

4.进程与线程

5.进程间通信

6.Socket与tcp/udp的关系，socket的组成

7.JMM模型，为什么线程有工作内存

8.事务的ACID特性，讲一下A和C

9.TCP三次握手，两次握手可以吗

10.讲一下拥塞控制

11.对链表进行Sort，Node的val为final，不允许使用数组，List，因为考察的是对链表的操作

12.非严格递增数组1，2，3，3，3，4，5，找到目标数字所在的第一个位置，无则输出-1，如3，输出2，要求O(log n);

13.反问，真难问，不知道问啥

虽然面得感觉一般般，还是许愿二面

作者：芒果味的周俊杰  
链接：[https://www.nowcoder.com/discuss/471145](https://www.nowcoder.com/discuss/471145" \t "_blank)  
来源：牛客网  
  
1.自我介绍   
2.实习项目，项目难点   
3.很久之前做的秒杀项目，问现在再来看这个项目，有哪些地方会有问题，改进，优化；   
4.进程和线程的区别，进程的通信方式   
4.Tcp和udp区别，应用场景   
5.DNS解析属于TCp还是udp，Dns解析过程   
6做题，判断两个字符串是否是相似字符串，相似的含义是只允许交换一次字符串的两个位置，使得和第二个字符串相等，那么他们就是相似的。   
面试官给的测试用例：   
相似字符串   
ab, ba => true   
abax, abax => true   
abc, abc => false   
aac, abc => false

作者：我爱offer爱我  
链接：[https://www.nowcoder.com/discuss/477948](https://www.nowcoder.com/discuss/477948" \t "_blank)  
来源：牛客网

面试官虽然带着口罩，但是真是一个帅气的小哥哥，人真好呢~

自我介绍+项目介绍

1数据库三大范式

2数据库的四种隔离级别，出现的问题

3触发器和数据库存储

1抽象类和接口

2finally

做题，两个字符串是否由相同字符构成。

莫名觉得面的很开心~~~

进线程区别   
调度算法   
协程   
虚拟内存   
udp为啥不可靠   
可靠udp怎么实现   
数据库索引   
数据库隔离级别   
算法：topK（快排思想，结果写了二十分钟才完事，都没时间出第二题）

作者：小Kakao加油，offer一定会有的  
链接：[https://www.nowcoder.com/discuss/478320](https://www.nowcoder.com/discuss/478320" \t "_blank)  
来源：牛客网

### 8月15日 一面

1. 问项目
2. cookie和session的区别？
3. Java的反射机制了解吗？
4. 反射机制中获得实例的方式有几种？分别是什么？
5. getClass和forName得到的Class实例是同一个实例还是不同的实例？
6. 讲一下ClassLoader？
7. 泛型了解吗？
8. 泛型中类型检查是在编译阶段还是运行阶段？
9. 泛型中，一个容器存的是String，我们add int类型能编译成功吗？
10. Java中的网络编程了解吗？
11. MySQL的索引是什么数据结构？
12. MySQL的query用大于(>)小于(<)的时候会用的索引吗？like呢？
13. 说一下left join， right join， inner join？
14. 知道哪些设计模式？
15. 算法 1：[二叉树](/jump/super-jump/word?word=二叉树" \t "_blank)的深度
16. 算法 2: 两个字符串的最长公共子串（太紧张，没写出来）

链接：[https://www.nowcoder.com/discuss/478124](https://www.nowcoder.com/discuss/478124" \t "_blank)

自我介绍

项目中遇到的技术难点

项目中遇到的问题，（方案讨论中吧 非技术上的）

网络：

TCP UDP的区别

TCP三次握手，四次挥手

为什么要四次挥手？

操作系统：

进程与线程的区别

算法：

给一个队列，借助两个栈，将队列中的元素[排序](/jump/super-jump/word?word=排序" \t "_blank)

字符串[排序](/jump/super-jump/word?word=排序" \t "_blank) 首先按照长度[排序](/jump/super-jump/word?word=排序" \t "_blank)，然后按照字符串[排序](/jump/super-jump/word?word=排序" \t "_blank)。I am a an student.------->I a am an student

作者：Samantha\_fate  
链接：[https://www.nowcoder.com/discuss/492731](https://www.nowcoder.com/discuss/492731" \t "_blank)  
来源：牛客网

聊天：

1、对后端的理解，为啥做后端，为啥觉得自己适合后端

项目：

1、项目

2、restful风格解释一下, restful原则还是规范来着忘了，不知道，queryparam和pathparam区别？总之就着restful往下问

3、get、post等区别

4、我项目中使用redis的场景，为什么要用redis，为什么不用数据库，redis具体比数据库快多少？

5、消息队列处理的时候，万一消费者处理失败怎么办

工程应用：

1、平时做项目有用过git么？问如果已经commit了某代码，现在你对改代码做了一点点修改，你又不想再全部commit一遍，咋整？？？

2、平时用linux么？如果想查看所有正在运行的java程序怎么做？

java：

1、String StringBuilder StringBuffer的区别，[阿里巴巴](/jump/super-jump/word?word=阿里巴巴" \t "_blank)编程规约里面有一条说不能在for循环体里写 str += "a"; 这种代码，为什么？

2、ArrayList和LinkedList的区别，增删改查效率

3、接口和抽象类的区别

代码：

[二叉树两个节点的最近公共祖先](/jump/super-jump/practice?questionId=1024325" \t "_blank)