|  |
| --- |
| 1又走出一步:  把自己的代码用git统一管理。这个管理是exercise仓库。 |
| <http://blog.csdn.net/u014079773/article/details/51595127>  这是一篇写的很不错的eclipse中使用git的文章。 |
| 下午做了一道求binary tree的题目，挺好的，看别人的代码，多练习，你就是牛人。每天把代码同步到github上。让自己看到自己的成长。 |
| 还需要做的是，参加一个很有规模的比赛，让自己真的有ML项目。 |
| 1．深度学习了深度优先和广度优先两种算法。现在都可以手动写出来，挺好的。  明天还需要加油的是，要提高编程的效率。 |

**一道题目搞定深度优先，还是广度优先，非常棒。（不行动，永远不知道自己到底是不是真的能够做到。）**

|  |
| --- |
| minimum-depth-of-binary-tree  Given a binary tree, find its minimum depth.The minimum depth is the number of nodes along the shortest path from the root node down to the nearest leaf node. |
| // 二叉树操作主要还是利用尾递归或者循环遍历这两种思路，进而涉及DFS（主要利用递归或者栈实现）或者BFS（主要利用队列实现）。剩下的只需要按照这些思路即可。 |
| 1. 用递归。也就是用深度优先。   **public** **int** run(TreeNode root) {  **if**(root==**null**) **return** 0;//出口  **if**(root.left==**null**) **return** 1+run(root.right);//只有左边，找到左边子树的深度。|出口  **if**(root.right==**null**) **return** 1+run(root.left);//只有右边，找到右子数的深度即可。|出口  **return** 1+Math.*min*(run(root.left), run(root.right));//两边都有，那么比较选择小的。  } |
| //方法1广度优先遍历，非递 归包含深度递归的算法  **public** **class** Solution1 {  **public** **int** run(TreeNode root) {  **if**(root==**null**) **return** 0;  Queue<TreeNode> que=**new** LinkedList<TreeNode>();  que.add(root);  **int** cnt=0;  **while**(!que.isEmpty())  { **int** size=que.size();  ++cnt;  **for**(**int** i=0;i<size;++i)  {  TreeNode cur=que.poll();  **if**(cur.left==**null**&&cur.right==**null**)  **return** cnt;  **if**(cur.left!=**null**)  que.add(cur.left);  **if**(cur.right!=**null**)  que.add(cur.right);    }  }  **return** cnt;  }  } |
| 用深度遍历。不用递归，用栈的做法。  就本题目而言，肯定是深度优先的效率更高。 |
| 记昨天的一道字节跳动的题目，讲的是。   1. userId deviceId这两个字段。有用户用过多个device登录，找出这样的圈子。   这个题目的思路昨天我第一种做法就是用递归去做，递归考虑问题，往往比较直接和方便。  不过以后要知道，其实那就是深度优先遍历。如果要深度优先遍历的话，那么还可以用栈结构来解决。  同样，如果用深度能解决的，应该用广度也能解决。广度遍历就不需要用到递归了。这个只需要用到队列这种结构就可以了，这样可以向面试官展示我们的数据结构功底。挺好的。以后的深度优先和广度优先就用这两道题做范例法。用范例驱动是最快的方式。Nice |
|  |
|  |

|  |
| --- |
| Mysql的四大隔离级别： |
| 百度运维一面：  网络：  1.tcp和udp的区别  2.tcp里边拥塞控制慢启动的原理，滑动窗口默认值？  3.OSI七层  4.HTTP协议的了解？1.0和2.0版本的区别？HTTP的状态码的了解  操作系统：  1.进程和线程的区别  2.产生死锁的四个必要条件  Linux：  1.查看Linux系统的一些命令  2.可以看到cpu的所有的命令，尽量说  3.ps结果从哪调用  4.du和df显示不一致，为什么？  Python  1.深拷贝和浅拷贝的区别  2.python如何复制一个对象  项目：  本科期间做的方向  研究生期间的课题研究方向  百度运维二面：  首先是直接项目介绍  1.介绍下项目？项目的背景？项目的描述？你在项目中做了什么？  2.根据你所说的，对你在项目中的描述开始问问题？（之后介绍项目时，最好能把自己的项目理一遍，写出来，说自己会的点，不要牵扯出自己不会的问题）  3.在项目中高可用那块的设计是怎样做的？实现的原理是什么？里边有说vip，然后那个vip实现的原理是什么？  4.zabbix的数据采集方式有哪些？有说SNMP协议，然后又问SNMP的优点有哪些？  5.zabbix的数据库为什么会有瓶颈？说的瓶颈体现在哪些？有没有好的优化方法？zabbix数据库索引是如何实现的?  6.过渡到数据库，平常做的数据库的优化有哪些？  7.索引内部的存储方式是什么样的？hash索引，Btree索引的区别，B-tree索引的优点有哪些？hash索引效率高是为什么，其数据结构？又扯到散列表，解决地址冲突等一些。  8.数据库的复制原理？其中SQL线程的回放是什么？  9.对于数据库方面的备份是怎么考虑的?  网络方面：  1.TCP和UDP的区别  2.TCP保障连接可靠的方法有哪些？  3.SYN攻击，解决办法  操作系统方面  1.死锁的产生？死锁的解决办法？  2.进程是如何切换的？在进程处于就绪态时还会有发生哪些事情？进程的状态  3.数据从内存写到磁盘上发生的过程，具体行为是什么？  Linux  1.vfs，ext2,ext3,ext4的区别  2.文件系统  设计题  海量IP数据的存储，两个文件，合到一起，去掉重复的，怎么设计？    教训：  谈项目时，自己不熟的那个点，不要着重讲，会让面试官误认为你很熟，谈的不好，会很尴尬  **简历上写的熟悉技能最好不要太范** |
| **蘑菇街：中石油**  面完第三天发现已挂，原因应该是基础太差，不过看了这么久牛客网，觉得时候回报下了，赶紧建个号来发个面经攒攒rp，记录下踩过的坑。（为了尽量还原真实场景。。用对话方式写的。。。不喜勿喷）  面试在上午10点，总计35分钟，我是带着耳机跑到大街上去面试的\笑哭  1. 自我介绍balala~~由于自己成绩还不错也拿过几个国奖于是面试官说我简历挺丰富的（又在给自己挖坑了。。。。）  面试官：我这有个简单的问题（\手动滑稽）  我：。。。。（坑一般都是这么挖的）  面试官：你先给我说说链表的数据结构吧  我：链表是线性表的一种，有单向的、双向的和循环链表。  面试官：不对啊，我是问的数据结构  我：（我擦？？）链表的节点一般由2个部分构成：数据域、指向下一个节点的指针  面试官：好的，现在有个简单问题，要把一个单向链表反转下，你讲讲思路吧，最好拿支笔画画想清楚再说  我：（在街上哪来的笔。。）好的，等我一分钟。。（\笑哭原地蹲下拿手比划下）  （一分钟后。。）  我：先定义一个函数传入头结点指针（头结点是保存数据的）返回值是节点指针，如果传入头结点指针为null则返回null；否则开始循环遍历每一个节点；每遇到一个节点，先保存他的下一节点和下下一个节点，如果下一节点为null则反转完成，否则将下一节点的next指向当前节点；再将下一节点赋予用于循环的变量，再把下下一节点付给下一节点的变量。。。如此进行下去。。  面试官：时间和空间复杂度？  我：时间O（N），空间O（1）  面试官：好的，思路基本正确。我看你简历上写着有个项目遇到了异地部署的问题，你是怎么解决的？  我：（一面就问项目了？？？）一台主机在公网，一台在内网，通过mysql主从复制的方式，在公网上部了台写库，在内网部了台读库，之所以不直接在公网上部mysql是因为内网的server在访问数据库时总要三次握手建立连接，这会导致每次ajax请求带有80~200毫秒的延迟。  面试官：那用户要想导入数据，得去公网？  我：（excuse me？？？）读写的分离是在server层做的，通过区分写的sql语句和读的sql语句来送到不同的mysql去执行。  **面试官：哦，好，你给说说数据库的事务隔离级别吧。**  我：（不问三次握手了？？）有四种：一是未提交读，二是提交读，三嘛（我擦突然忘记了，这东西貌似会导致脏读），四是可串行化。。。。  面试官：第三个是**可重复读**，你mysql还得好好看啊。  我：（形式不妙，感觉要开始跳坑了）是的。。是的。。。  面试官：我看你的实习上写着用到了支付模块，是怎么做的？  我：接入了微信平台  面试官：噢，那说说这个流程吧。  我：接入微信平台时有两个，一个是开放平台，一个是公众号，由于支付需要用到openid，所以得先从第三方登录讲起。  面试官：噢，那你知道微信是怎么做第三方登录的吗，是用到了什么协议？  我：oAuth协议。  面试官：这个协议里面有个code他全称叫啥？  我：au。。。authorization\_code ？  面试官：是授权码  我：（我擦。。。中文我也会啊）。。。噢噢是的是的  面试官：。。。你讲讲这个授权的过程吧  我：这个协议要运行起来得有三方，一是用户，二是授权服务器，三是第三方，也就是我们的server；授权发起是由我们自行拼接授权页面的链接，带上自己的回调url，让用户访问这个链接；用户进入链接，跟授权服务器验证信息，验证完后授权服务器会将授权码回调给我们的server，我们拿这个code去换取access\_token，之后就能拿到用户的信息了，再之后就是微信用户入自己的数据库的问题了，这里牵涉到openid和公众号/应用的绑定关系，转换时需要用unionid进行转换。。。（突然打断，本来想继续往下说在转换的时候遇到的坑）  面试官：好了，这个授权没这么复杂吧。。  我：。。。。  面试官：这个授权其实很简单，其实就是第一步拿code，第二步用code换access\_token，第三步拿access\_token换用户信息，你在面试后可以自己去总结下，没这么复杂的。  我：（。。。我这不是怕你听不懂么）好的。。好的。。  面试官：好了，今天面试就到这吧，你有啥问题想问吗？  我：（不讲支付了？？？）。。。。（估计是不想面了吧\笑哭）  我之后问了三个问题，是关于平台语言和数据库选型相关的，之后就说hr会通知我后续的安排。  于是等啊等啊等到自己去官网查下发现已经挂了，秋招第一面就gg了，得好好打基础了\笑哭 |
| 一面：  数据倾斜怎么处理？  写一个算法判断一个图是不是DAG？（答案是拓扑排序，我一直在说bfs，尴尬，这边花了太多时间，O\_\_O "…）  flume用过是吧？flume数据源如何监听文件夹里有新文件产生？  用过实时计算吗？没用过  mapreduce的map进程和reducer进程的jvm垃圾回收器怎么选择可以提高吞吐量？  二面：  会Spark吗？不会（面试官很意外）  用过Hive是吧？写HiveQL：两张表：一张歌曲表，一张专辑表，找出每张专辑的Top100。（这边花了太多时间，写出来后有2处不对，group by和select的字段不一致，join语法不对(╯﹏╰)）  HBase用过是吧？画一下它的架构图  知道HBase的LSM结构吗？乱说一通后，那我这边问完了，出去等通知吧！  HR面：  想留在杭州吗？父母对你留杭州同意吗？期待的薪资多少？还拿过哪些offer？  （个人感觉2面的面试官想问spark，可是因为我不会，所以聊的时间比较短，对spark不熟的童鞋可以参考一下我的面经，  可能有点短，前面的时间主要在聊之前做的项目和实习经历） |
| **链家 中石油**  **进去先写一个小时代码，五道题，我写了三道代码和两道思路。** 一面， 自我介绍， (巴拉巴拉，学校，项目，科研) 最满意的项目，为什么满意， (巴拉巴拉，科研结合实践，项目从头跟到尾，心目中遇到问题的解决方法) 讲一下聚类，初始聚类中心怎么选，选几个 (聚类原理，用经验选，降维选，有论文用ICA这些只能算法求) 讲一下hadoop那套 （hdfs,mapreduce,shuffle还给他画上图了） 讲一下svm推导 （手推到对偶形式然后说了说线性可分不可分和几种核函数效果） svm和lr的区别 （最大化间隔平面和最大似然估计） 你觉得你做的技术怎么用在链家里？ （做分类，预测新用户感兴趣的房子啊什么的） 你有什么问题吗？ 我问链家数据挖掘团队 面试官说有三十来人，主要用聚类和分类。   。。。。。。。。 等结果，五分钟后显示挂了，我去问面试官哪里做的不好，面试官说**代码写的不行**，我就回去收拾走人了。 面试官又出来和我说去和负责人说说，觉得我算法功底很好，希望给个二面。 于是继续等，两三分钟后来和我说负责人觉得我代码功底还要提高，再接再厉。   我想说，你们觉得我代码不好干嘛还让我一面，你们想问啥算法问题我都能给你们说上来了，你们又从代码方面来拒我，可笑，面试官还是很好的 |
| spark源码提交了多少行  数据倾斜，怎么定位、怎么解决（这个问题挺重要的，阿里三轮技术面都被问了）  mapreduce过程  spark shuffle（三种），各种使用的场景，优势  spark排序实现流程，reduce端怎么实现的  hashpartitioner与rangePartitioner的实现，顺带讲了一下水塘抽样  spark有哪几种join，使用场景，以及实现原理  画一个yarn架构图，讲一下通信流程  hdfs数据读取流程，实现原理  就记得这些了  网易二面  先说了之前提交的spark源码  谈了一下项目  dagschedule、taskschedule、schedulebankend实现原理  spark join实现  宽依赖、窄依赖  用scala写了一个word count，然后一个sql场景题目  jvm模型，gc算法  其他的记不得了，和这个面试关主要在谈spark源码  hr面  谈谈学校、爱好、自我评价。。。最后问有多少hc。。。 |
|  |
|  |

**整理一下计算机网络部分的面试常考点，参考书籍：《计算机网络》第五版 谢希仁的那本，希望对大家有所帮助**

**OSI，TCP/IP，五层协议的体系结构，以及各层协议**

**OSI分层 （7层）**：物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。

**TCP/IP分层（4层）**：网络接口层、 网际层、运输层、 应用层。

**五层协议     （5层）**：物理层、数据链路层、网络层、运输层、 应用层。

**每一层的协议如下：**

物理层：RJ45、CLOCK、IEEE802.3    （中继器，集线器，网关）

数据链路：PPP、FR、HDLC、VLAN、MAC  （网桥，交换机）

网络层：IP、ICMP、ARP、RARP、OSPF、IPX、RIP、IGRP、 （路由器）

传输层：TCP、UDP、SPX

会话层：NFS、SQL、NETBIOS、RPC

表示层：JPEG、MPEG、ASII

应用层：FTP、DNS、Telnet、SMTP、HTTP、WWW、NFS

**每一层的作用如下：**

**物理层：**通过媒介传输比特,确定机械及电气规范（比特Bit）

**数据链路层**：将比特组装成帧和点到点的传递（帧Frame）

**网络层**：负责数据包从源到宿的传递和网际互连（包PackeT）

**传输层**：提供端到端的可靠报文传递和错误恢复（段Segment）

**会话层**：建立、管理和终止会话（会话协议数据单元SPDU）

**表示层**：对数据进行翻译、加密和压缩（表示协议数据单元PPDU）

**应用层**：允许访问OSI环境的手段（应用协议数据单元APDU）

**IP地址的分类**

A类地址：以0开头， 第一个字节范围：0~127（1.0.0.0 - 126.255.255.255）；

B类地址：以10开头，    第一个字节范围：128~191（128.0.0.0 - 191.255.255.255）；

C类地址：以110开头，  第一个字节范围：192~223（192.0.0.0 - 223.255.255.255）；

10.0.0.0—10.255.255.255， 172.16.0.0—172.31.255.255， 192.168.0.0—192.168.255.255。（Internet上保留地址用于内部）

**IP地址与子网掩码相与得到主机号**

**ARP是地址解析协议，简单语言解释一下工作原理。**

1：首先，每个主机都会在自己的ARP缓冲区中建立一个ARP列表，以表示IP地址和MAC地址之间的对应关系。

2：当源主机要发送数据时，首先检查ARP列表中是否有对应IP地址的目的主机的MAC地址，如果有，则直接发送数据，如果没有，就向本网段的所有主机发送ARP数据包，该数据包包括的内容有：**源主机** **IP地址，源主机MAC地址，目的主机的IP 地址**。

3：当本网络的所有主机收到该ARP数据包时，首先检查数据包中的IP地址是否是自己的IP地址，如果不是，则忽略该数据包，如果是，则首先从数据包中取出源主机的IP和MAC地址写入到ARP列表中，如果已经存在，则覆盖，然后将自己的MAC地址写入ARP响应包中，告诉源主机自己是它想要找的MAC地址。

4：源主机收到ARP响应包后。将目的主机的IP和MAC地址写入ARP列表，并利用此信息发送数据。如果源主机一直没有收到ARP响应数据包，表示ARP查询失败。

广播发送ARP请求，单播发送ARP响应。

**各种协议**

**ICMP协议：** 因特网控制报文协议。它是TCP/IP协议族的一个子协议，用于在IP主机、路由器之间传递控制消息。

**TFTP协议：** 是TCP/IP协议族中的一个用来在客户机与服务器之间进行简单文件传输的协议，提供不复杂、开销不大的文件传输服务。

**HTTP协议：** 超文本传输协议，是一个属于应用层的面向对象的协议，由于其简捷、快速的方式，适用于分布式超媒体信息系统。

**DHCP协议：** 动态主机配置协议，是一种让系统得以连接到网络上，并获取所需要的配置参数手段。

**NAT协议**：网络地址转换属接入广域网(WAN)技术，是一种将私有（保留）地址转化为合法IP地址的转换技术，

**DHCP协议：**一个局域网的网络协议，使用UDP协议工作，用途：给内部网络或网络服务供应商自动分配IP地址，给用户或者内部网络管理员作为对所有计算机作中央管理的手段。

**描述：RARP**

RARP是逆地址解析协议，作用是完成硬件地址到IP地址的映射，主要用于无盘工作站，因为给无盘工作站配置的IP地址不能保存。工作流程：在网络中配置一台RARP服务器，里面保存着IP地址和MAC地址的映射关系，当无盘工作站启动后，就封装一个RARP数据包，里面有其MAC地址，然后广播到网络上去，当服务器收到请求包后，就查找对应的MAC地址的IP地址装入响应报文中发回给请求者。因为需要广播请求报文，因此RARP只能用于具有广播能力的网络。

**TCP三次握手和四次挥手的全过程**

**三次握手：**

第一次握手：客户端发送syn包(syn=x)到服务器，并进入SYN\_SEND状态，等待服务器确认；

第二次握手：服务器收到syn包，必须确认客户的SYN（ack=x+1），同时自己也发送一个SYN包（syn=y），即SYN+ACK包，此时服务器进入SYN\_RECV状态；

第三次握手：客户端收到服务器的SYN＋ACK包，向服务器发送确认包ACK(ack=y+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。

握手过程中传送的包里不包含数据，三次握手完毕后，客户端与服务器才正式开始传送数据。理想状态下，TCP连接一旦建立，在通信双方中的任何一方主动关闭连接之前，TCP 连接都将被一直保持下去。

**四次握手**

与建立连接的“三次握手”类似，断开一个TCP连接则需要“四次握手”。

第一次挥手：主动关闭方发送一个FIN，用来关闭主动方到被动关闭方的数据传送，也就是主动关闭方告诉被动关闭方：我已经不 会再给你发数据了(当然，在fin包之前发送出去的数据，如果没有收到对应的ack确认报文，主动关闭方依然会重发这些数据)，但是，此时主动关闭方还可 以接受数据。

第二次挥手：被动关闭方收到FIN包后，发送一个ACK给对方，确认序号为收到序号+1（与SYN相同，一个FIN占用一个序号）。  
第三次挥手：被动关闭方发送一个FIN，用来关闭被动关闭方到主动关闭方的数据传送，也就是告诉主动关闭方，我的数据也发送完了，不会再给你发数据了。  
第四次挥手：主动关闭方收到FIN后，发送一个ACK给被动关闭方，确认序号为收到序号+1，至此，完成四次挥手。



**在浏览器中输入www.baidu.com后执行的全部过程**

1、客户端浏览器通过DNS解析到www.baidu.com的IP地址220.181.27.48，通过这个IP地址找到客户端到服务器的路径。客户端浏览器发起一个HTTP会话到220.161.27.48，然后通过TCP进行封装数据包，输入到网络层。

2、在客户端的传输层，把HTTP会话请求分成报文段，添加源和目的端口，如服务器使用80端口监听客户端的请求，客户端由系统随机选择一个端口如5000，与服务器进行交换，服务器把相应的请求返回给客户端的5000端口。然后使用IP层的IP地址查找目的端。

3、客户端的网络层不用关系应用层或者传输层的东西，主要做的是通过查找路由表确定如何到达服务器，期间可能经过多个路由器，这些都是由路由器来完成的工作，我不作过多的描述，无非就是通过查找路由表决定通过那个路径到达服务器。

4、客户端的链路层，包通过链路层发送到路由器，通过邻居协议查找给定IP地址的MAC地址，然后发送ARP请求查找目的地址，如果得到回应后就可以使用ARP的请求应答交换的IP数据包现在就可以传输了，然后发送IP数据包到达服务器的地址。

**TCP和UDP的区别？**

TCP提供面向连接的、可靠的数据流传输，而UDP提供的是非面向连接的、不可靠的数据流传输。

TCP传输单位称为TCP报文段，UDP传输单位称为用户数据报。

TCP注重数据安全性，UDP数据传输快，因为不需要连接等待，少了许多操作，但是其安全性却一般。

**TCP对应的协议和UDP对应的协议**

**TCP对应的协议：**

（1） **FTP**：定义了文件传输协议，使用**21**端口。

（2） **Telnet**：一种用于远程登陆的端口，使用**23**端口，用户可以以自己的身份远程连接到计算机上，可提供基于DOS模式下的通信服务。

（3） **SMTP**：邮件传送协议，用于发送邮件。服务器开放的是**25**号端口。

（4） **POP3**：它是和SMTP对应，POP3用于接收邮件。POP3协议所用的是**110**端口。

（5）**HTTP**：是从Web服务器传输超文本到本地浏览器的传送协议。

**UDP对应的协议：**

（1） **DNS**：用于域名解析服务，将域名地址转换为IP地址。DNS用的是**53**号端口。

（2） **SNMP**：简单网络管理协议，使用**161**号端口，是用来管理网络设备的。由于网络设备很多，无连接的服务就体现出其优势。

（3）**TFTP**(Trival File Tran敏感词er Protocal)，简单文件传输协议，该协议在熟知端口**69**上使用UDP服务。

**DNS域名系统，简单描述其工作原理。**

当DNS客户机需要在程序中使用名称时，它会查询DNS服务器来解析该名称。客户机发送的每条查询信息包括三条信息：包括：指定的DNS域名，指定的查询类型，DNS域名的指定类别。基于UDP服务，端口53. 该应用一般不直接为用户使用，而是为其他应用服务，如HTTP，SMTP等在其中需要完成主机名到IP地址的转换。

**面向连接和非面向连接的服务的特点是什么？**

面向连接的服务，通信双方在进行通信之前，要先在双方建立起一个完整的可以彼此沟通的通道，在通信过程中，整个连接的情况一直可以被实时地监控和管理。

非面向连接的服务，不需要预先建立一个联络两个通信节点的连接，需要通信的时候，发送节点就可以往网络上发送信息，让信息自主地在网络上去传，一般在传输的过程中不再加以监控。

**TCP的三次握手过程？为什么会采用三次握手，若采用二次握手可以吗？**

答：建立连接的过程是利用客户服务器模式，假设主机A为客户端，主机B为服务器端。

（1）TCP的三次握手过程：主机A向B发送连接请求；主机B对收到的主机A的报文段进行确认；主机A再次对主机B的确认进行确认。

（2）采用三次握手是为了防止失效的连接请求报文段突然又传送到主机B，因而产生错误。失效的连接请求报文段是指：主机A发出的连接请求没有收到主机B的确认，于是经过一段时间后，主机A又重新向主机B发送连接请求，且建立成功，顺序完成数据传输。考虑这样一种特殊情况，主机A第一次发送的连接请求并没有丢失，而是因为网络节点导致延迟达到主机B，主机B以为是主机A又发起的新连接，于是主机B同意连接，并向主机A发回确认，但是此时主机A根本不会理会，主机B就一直在等待主机A发送数据，导致主机B的资源浪费。

（3）采用两次握手不行，原因就是上面说的实效的连接请求的特殊情况。

**端口及对应的服务？**

|  |  |  |  |
| --- | --- | --- | --- |
| 服务 | 端口号 | 服务 | 端口号 |
| FTP | 21 | SSH | 22 |
| telnet | 23 | SMTP | 25 |
| Domain(域名服务器) | 53 | HTTP | 80 |
| POP3 | 110 | NTP（网络时间协议） | 123 |
| MySQL数据库服务 | 3306 | Shell或 cmd | 514 |
| POP-2 | 109 | SQL Server | 1433 |

**IP数据包的格式**

IP数据报由**首部** 和**数据**  两部分组成。首部由**固定部分和可选部分**  组成。**首部的固定部分有** **20 字节**。可选部分的长度变化范围为1——40字节。固定部分的字段：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 位数（bit） | 字段名 | 位数 |
| 版本 | 4  Ipv4 | 首部长度 | 4（表示的最大数为15个单位，一个单位表示4字节） |
| 服务类型 | 8  以前很少用 | 总长度 | 16 （首部和数据部分的总长度，因此数据报的最大长度为65535字节，即64KB，但是由于链路层的MAC都有一定的最大传输单元，因此IP数据报的长度一般都不会有理论上的那么大，如果超出了MAC的最大单元就会进行分片） |
| 标识 | 16 （相同的标识使得分片后的数据报片能正确的重装成原来的数据报） | 标志 | 3 （最低位MF=1表示后面还有分片，MF=0表示这是若干个数据报片的最后一个中间位DF=0才允许分片） |
| 片偏移 | 片偏移指出较长的分组在分片后，某片在原分组中的相对位置，都是8字节的偏移位置 | 生存时间 | 数据报在网络中的生存时间，指最多经过路由器的跳数 |
| 协议 | 8 （指出该数据报携带的数据是何种协议，以使得目的主机的IP层知道应将数据部分上交给哪个处理程序）如ICMP=1 IGMP=2 TCP=6 EGP=8 IGP=9 UDP=17 Ipv6=41 OSPF=89 | 首部校验和 | 这个部分只校验首部，不包括数据部分，计算方法：将首部划分为多个16位的部分，然后每个16位部分取反，然后计算和，再将和取反放到首部校验和。接收方收到后按同样的方法划分，取反，求和，在取反，如果结果为零，则接收，否则就丢弃 |
| 源地址 | 32 | 目的地址 | 32 |

**TCP数据报的格式？**

一个TCP报文段分为首部和数据两部分。首部由固定部分和选项部分组成，固定部分是20字节。TCP首部的最大长度为60。首部固定部分字段：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字节（Byte） | 字段名 | 字节（Byte） |
| 源端口 | 2 | 目的端口 | 2 |
| 序号 | 4 | 确认号 | 4，是期望收到对方的下一个报文段的数据的第一个字节的序号 |
| 数据偏移 | 4bit 指出TCP报文段的数据起始处距离TCP报文段的起始有多远 | 保留 | 6bit |
| 紧急比特URG |  | 确认比特ACK | 只有当ACK=1时，确认号字段才有效 |
| 推送比特PSH |  | 复位比特RST |  |
| 同步比特SYN |  | 终止比特FIN |  |
| 窗口 | 2 | 检验和 | 2 （包括首部和数据两部分，同时还要加12字节的伪首部进行校验和计算） |
| 选项 | 长度可变（范围1——40） |  |  |

**TCP的12字节伪首部：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 源IP地址（4） | 目的IP地址（4） | 0 (1) | 6(1) 代表这是TCP，IP协议中提到过 | TCP长度（2） |

**TCP数据报的格式？**

用户数据报UDP由首部和数据部分组成。首部只有8个字节，由4个字段组成，每个字段都是两个字节。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字节 | 字段名 | 字节 |
| 源端口 | 2 | 目的端口 | 2 |
| 长度 | 2 | 检验和 | 2 （检验首部和数据，加12字节的伪首部） |

**UDP的12字节伪首部：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 源IP地址（4） | 目的IP地址（4） | 0 (1) | 17(1) 代表这是UDP | UDP长度（2） |

**以太网MAC帧格式？**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 前导码 | 前定界符 | 目的地址 | 源目的地址 | 长度字段 | 数据字段 | 校验字段 |
| 7B | 1B | 6B | 6B | 2B | 46-1500 | 4B |

**了解交换机、路由器、网关的概念，并知道各自的用途**

1）交换机

在计算机网络系统中，交换机是针对共享工作模式的弱点而推出的。交换机拥有一条高带宽的背部总线和内部交换矩阵。交换机的所有的端口都挂接在这条背 部总线上，当控制电路收到数据包以后，处理端口会查找内存中的地址对照表以确定目的MAC（网卡的硬件地址）的NIC（网卡）挂接在哪个端口上，通过内部 交换矩阵迅速将数据包传送到目的端口。目的MAC若不存在，交换机才广播到所有的端口，接收端口回应后交换机会“学习”新的地址，并把它添加入内部地址表 中。

交换机工作于OSI参考模型的第二层，即数据链路层。交换机内部的CPU会在每个端口成功连接时，通过ARP协议学习它的MAC地址，保存成一张 ARP表。在今后的通讯中，发往该MAC地址的数据包将仅送往其对应的端口，而不是所有的端口。因此，交换机可用于划分数据链路层广播，即冲突域；但它不 能划分网络层广播，即广播域。

交换机被广泛应用于二层网络交换，俗称“二层交换机”。

交换机的种类有：二层交换机、三层交换机、四层交换机、七层交换机分别工作在OSI七层模型中的第二层、第三层、第四层盒第七层，并因此而得名。

2）路由器

**路由器**（**Router**）是一种计算机网络设备，提供了路由与转送两种重要机制，可以决定数据包从来源端到目的端所经过 的路由路径（host到host之间的传输路径），这个过程称为路由；将路由器输入端的数据包移送至适当的路由器输出端(在路由器内部进行)，这称为转 送。路由工作在OSI模型的第三层——即网络层，例如网际协议。

**路由器的一个作用是连通不同的网络，另一个作用是选择信息传送的线路。** 路由器与交换器的差别，路由器是属于OSI第三层的产品，交换器是OSI第二层的产品(这里特指二层交换机)。

3）网关

**网关**（Gateway），**网关**顾名思义就是连接两个网络的设备，区别于路由器（由于历史的原因，许多有关TCP/IP 的文献曾经把网络层使用的路由器（Router）称为网关，在今天很多局域网采用都是路由来接入网络，因此现在通常指的网关就是路由器的IP），经常在家 庭中或者小型企业网络中使用，用于连接局域网和Internet。 网关也经常指把一种协议转成另一种协议的设备，比如语音网关。

在传统TCP/IP术语中，网络设备只分成两种，一种为网关（gateway），另一种为主机（host）。网关能在网络间转递数据包，但主机不能 转送数据包。在主机（又称终端系统，end system）中，数据包需经过TCP/IP四层协议处理，但是在网关（又称中介系 统，intermediate system）只需要到达网际层（Internet layer），决定路径之后就可以转送。在当时，网关 （gateway）与路由器（router）还没有区别。

在现代网络术语中，网关（gateway）与路由器（router）的定义不同。网关（gateway）能在不同协议间移动数据，而路由器（router）是在不同网络间移动数据，相当于传统所说的IP网关（IP gateway）。

**网关**是连接两个网络的设备，对于语音网关来说，他可以连接PSTN网络和以太网，这就相当于VOIP，把不同电话中的模拟信号通过网关而转换成数字信号，而且加入协议再去传输。在到了接收端的时候再通过**网关**还原成模拟的电话信号，最后才能在电话机上听到。

对于以太网中的**网关**只能转发三层以上数据包，这一点和路由是一样的。而不同的是**网关**中并没有路由表，他只能按照预先设定的不同网段来进行转发。网关最重要的一点就是端口映射，子网内用户在外网看来只是外网的IP地址对应着不同的端口，这样看来就会保护子网内的用户。

8.26

|  |
| --- |
| **链家**  下午约的的四点，提前过去就直接做笔试了。五道编程题，**第一道是剑指offer原题**，第二括号匹配，第三是环链表找到最小包含全部颜色的链表长度，第四是找出一段时间的恶意ip，第五个是找出模块依赖是否有循环依赖。 一面： 1、介绍下自己。 2、介绍kafka架构 3、介绍spark提交作业运行流程 4、MR过程 5、机器学习，如何理解模型是什么。 6、推导最熟悉的机器学习算法 7、workcount手写代码 8、RDD内部数据结构 9、为什么不用Hive和SparkSQL查询？ 10、和面试官讨论了一会笔试题第四题如何实现。 11、本来想问第五题，看我写了就没问，以为我没做出来。。 12、有什么想问他的 二面 1、看你简历写的熟悉Java.那我就问你几个Java的知识吧，**垃圾回收的结构。(新生代，老年代，正说着新生代的垃圾回收算法就把我打断了，下一题)** 2、给你出道题吧，你用Java写个函数。有百万条数据，每行有三列，没列用-隔开第一行可以是abc-123-hsgf，我们假设两条数据中，对应的每列中，若包含的字符个数和种类一样顺序可不一样，将其归为一类。输出每个类有多少条数据。最开始没理解，然后面试官将数据降为一列。😂我说可以用Spark写么，他说可以，说了个什么用Scala也要写函数，我说我用java写，他说那你写吧。。。写的时候我将每行转为字符数组，然后进行排序，因为同类的不管顺序如何，排了以后，顺序会一样的，然后将其放入一个全局的map中，统计几天。(后来想想也就是变相的wordcount嘛) 3、yarn是什么？ 4、yarn和spark的关系 5、spark提交到yarn上以后，具体发生了什么？ 6、如果一个节点挂掉以后会怎样？ 7、失败的任务，会丢失数据么，为什么？不会，为什么不会？ 8、hadoop架构 9、如果一个新人从根节点删除了HDFS目录，能恢复么？如何恢复？(本来答得HDFS的secondnamenode这方面，面试官说那我中间隔了很久了呢？好像是不能恢复了吧，我说，不太确定。。) 10、根据自我评价问，如果给你分配了一个你没接触过得任务，你怎么做？ 11、你怎么证明你说的，你是一个有责任心的人？ 三面:HR面 最想去的公司？ 为什么选链家？其他的为什么不选？ 你了解链家吗？ 你期望的公司文化是什么？ 你希望你的leader是什么样的人？ 你最早什么时候能来实习？ 结合两个场景，问怎么去处理。感觉像是在看性格。 感觉最虚的是HR面，问的一脸懵逼，感觉被看的透透的。在等候的时候，谈论到昨天面试卡的可能比较紧。我感觉面试就像判卷子。前期判的比较紧，判的多了要求可能会相对松一点吧。第一次发面经，攒攒运气。 |
| 我是通过**实习生转正**拿到的美团点评offer，岗位为：美团外卖结算组的**后端开发工程师（Java）。**  **实习生面试**  美团点评的实习生招聘比较喜欢内推，没有网申和笔试的环节，这一点大家要注意。2016年的5月号，我听到在大众点评工作的同学说美团点评开始招聘暑期实习生了，我查看了一下美团点评各个业务线主要负责的内容，加上同学的建议，最后选择了美团外卖事业部。  **一面**   * 在5月12号中午，我接到美团一面面试官预约面试的电话，开始面试官以为我在北京，当时我刚回到青岛，于是面试官提出帮我询问能不能电话面试，5月13号面试官打来电话说可以电话面试，并预约了电话面试的时间：**14号下午六点半**。 * 14号下午五点的时候面试官又打电话提醒了一次，六点半面试正式开始。 * 套路：先**自我介绍**：学历背景和项目经历。 * hashtable和hashmap的区别， hashmap底层的实现机制：负载因子、冲突解决方式。Java 8的优化。 * string 和stringbuffer的区别。 * JVM内存模型，每个区域存放的内容。 * JVM垃圾回收算法。 * JVM类加载机制、双亲委托模型。 * 两道算法题： 在一百个数中寻找重复数字，并对自己提出的算法进行更进一步的优化；判断在一个链表中是否有环。 * 多线程相关知识：synchronized关键字、Lock，并解释它们之间的区别。 * 常见的设计模式如工厂模式，自己看的代码中哪些地方用到了工厂设计模式。 * 线程池的优点和实现机制。 * 网络基础知识：TCP三次握手、四次挥手，为什么建立连接是三次，而断开连接要四次？ * HTTP POST 和 GET的区别和联系。 * 进程和线程的区别。 * 操作系统内存管理。 * 数据库知识，事务的四个特性ACID。 * 常见索引的实现原理，底层的数据结构B+树。 * 读过哪些源码，最近在看什么书？对MySQL 5.7等更新版本的关注。 * 有没有想问的问题。   总体来说一面的问题并不是，但是面试官特别喜欢引申，一个问题问完了往往会往深处探讨。这个大家要注意。  **二面**  在5月19号我接到了二面面试官的电话。二面面试官详细、着重询问问了我在15年的一段实习经历   * 实习过程中的任务、技术点。 * 实习公司的管理方式。 * 实习过程中最大的收获。 * 考察了个人对工作的态度和沟通能力。 * 一面面试官反馈了MySQL掌握的不错，于是二面面试官着重询问了MySQL相关知识。 * 嵌套查询、关联查询等常用SQL语句的书写。 * 常见慢查询优化的方式。 * 常见索引，innodb和myISAM的区别，innodb底层的实现机制，索引前缀的概念，B+树数据结构，为什么常见索引采用B+树的数据结构而不是平衡二叉树。 * 给出具体的业务场景，分析数据库表索引的设计。 * 对美团有没有想了解的。   **HR面**  5月30号接到了HR的电话，询问具体的实习时间，就发放了实习offer。  **公司实习的经历**  我是6月20号入职的，   * 早上去公司南京厅签实习合同。 * 美团实习生实行导师制，签完实习合同后由导师带领去资产处领取电脑（Mac Pro）,美团的研发岗位包括实习生，公司统一配备MAC。 * 找到自己的组，然后分配工位，就正式开始工作了。 * 前几天一般都是搭开发环境，熟悉工作环境，几天后组长就开始分配任务了。 * 我们组每天十点半开早会，十点半之前要到公司，组长一般九点四十五到。 * 组内中午有午休的习惯，十二点吃中午饭，下午两点开始工作，下午六点吃晚饭。 * 晚上八点之后下班有三十块钱的餐补，九点半之后下班打车报销。 * 有望京西、惠新西街南口、东湖渠三个班车点。 * 美团内部的技术分享特别多，每天都有，有公司级别的分享也有部门内部的分享，老大非常鼓励去听分享。 * 每天早会、两周一个迭代会议、三个月一次季度总结，美团的节奏还是比较快的。 * 美团在技术管理上给我的震撼特别大，我们都听惯了大公司内部撕逼的段子，然而在美团，最起码在我们组是不存在这种情况的，每个人都对自己的业务负责，实习期间我也跟随着同事经历了几次线上事故，出了问题大家的第一个反应是互相帮助、尽快解决线上问题，而不是相互推卸责任，在事故处理结束后，实事求是的分析事故原因，相关责任人做事故报告，并和大家探讨如何避免相同事故发生的方案。 * 另外一个,组长在分配任务的时候会考虑每一个的成长，根据每个人负责的任务合理分配业务开发和技术开发任务。 * 能人异士比较多，隔壁组有一个90后的T3-2(级别），我们组一个工作了好多年的技术专家界别才是T3-1。 * 美团点评是双休，总体上差不多是9 9 5的节奏。 * 大约每两个月有一次团队建设，出去漂流、骑马什么的。 * 恒电大厦C座下面新开了个食堂，一顿中午饭在16~20之间，相对来说还是能接受的，没什么特别好吃的档口，也可以选择去对面的摩托大厦或者隔壁锐创大厦吃饭，在摩托大厦吃饭还能碰见锤子团队，是的，就是罗永浩老师的锤子手机团队。   我觉得实习过程中，沟通特别重要，一定要做好风险前置，项目中遇到问题要及时反馈，避免要为自己而影响整个项目组的进程，项目delay的代价比较高，可能会影响个人的绩效。  **实习生转正面试**  美团点评在5月份正式招聘的这一批实习生是有转正机会的，需要通过转正面试。   * 在8月中旬校招组出台了实习生转正方案：转正窗口期有两次，分别是8月20号和9月12号，利用这两个窗口期来计算工作期限，针对Ｂ类实习生（5月份招聘的）需要组长评价，给出评定了的等级，分Ａ、Ｂ、Ｃ三个档次，评级Ａ的可以直接进入面试环节，评级为Ｂ的需要工作期限满8周才能进入面试环节，评级为Ｃ则被淘汰。 * 举个例子，我是6月20号入职的，到8月20号刚好满八周的工作期限，只要我的评级为Ｂ或者Ａ，那么我就可以进入第一个窗口期的面试，如果我是7月4号入职，评级为Ｂ，那我只能等第二个窗口期9月12号的时候再发起面试申请。 * 转正面试也分两次，第一次是组长面试，组长需要添一个评价表。第二次是部门老大面试。 * 组长面试内容：拉着导师坐一起，首先询问了我这段时间自己觉得做的最难的一个项目，最近哪一块做的好，哪一个部分做的不好，聊一下自己实习这段时间的收获，然后询问导师对我这段时间工作的评价。我个人觉得这个组长面试表现的不是很好，很多东西没有表达出来，没有表达出来自己对技术的热枕，没有对公司业务的深入探究。 * 部门老大面试：谈一谈自己实习这段时间经历的项目，分别描述一下，并针对项目内容进行一定的提问；谈一个自己觉得最难的项目，难在什么地方，最后怎么解决的。个人的教育经历：为什么不选择继续做算法，自己如何自学Java的。自己以后的职业规划。   转正面试过程持续了大约两个周，流程走完以后就是等待结果了，大约中秋前后接到了美团的HR邮件，通知转正面试通过了，正式offer发放是跟秋招一起发放的，在十月底左右发了正式offer，告知了薪资、福利等等，再然后就是邮寄三方，签约。 |
| 总结记录下自己的面试经历，大家共勉。  1. 笔试  笔试一共1个小时，5道编程题，现场手撸代码还是第一次，感觉还是时间还是有些紧张的。  1）给定一个数组，其中有一个数的出现次数超过数组长度的一半，要求找出这个数，空间复杂度和时间复杂度尽量低。  我的思路是对数组排序，那么要求的数就在排序后数组长度的一半处。这样没有使用额外空间，时间复杂度主要看排序的时间复杂度，我用的java.util.Arrays类中的Arrays.sort()方法，感觉是O(nlogn)  2）4类括号{}, [], (), <>，括号的出现规则是只能嵌套，不能交叉，例如{1,2}[a(b)]就是合法的，而{a)}就是不合法的。要求判断给定字符串中括号的使用是否合法。  我的思路是分别利用4个整型变量来记录4类括号的出现情况，遍历字符串，如果出现左括号，就将对应变量加1，如果出现右括号，就将对应的变量减一，如果这4个变量有任何一个小于0，那么就有不合法的括号，结束遍历。  3）一串珠子A，有m个珠子，n种不同颜色（n<m），要求子串B中包含所有的颜色，求子串B的长度最小是多少？  我用了双指针，i从前往后，j从后往前， i往后走，B为A中从i到j的子串，如果B中包含A中i处的珠子颜色，那么i继续往后走，直到B中不包含为止，然后j开始往前走，类似的，如果B中包含A中j处的珠子颜色，那么j继续往前走，否则停止。此时B的长度最小。  4）Web日志记录了IP和访问时间，恶意攻击指某一IP在时间t内（如10分钟内）连续访问的次数超过n次（如1024次），要求给定的Web日志中恶意攻击的IP地址。  这个题没来得及写，在面试的时候说了暴力的解法，用Map<IP, List<访问时间>>来存储，如果对应IP的List的size小于n，那么肯定不是恶意攻击，如果大于n，那么就从List的第一个元素开始遍历，元素的下标索引表示截止到该访问时间为止，一共有多少次访问。那么从头开始遍历，当遍历到第i个元素的时候，得到第(i+n)元素处的访问时间，如果两个时间相隔大于t，那么继续遍历，否则该IP存在恶意攻击。然后面试官说有两个问题：① 时间复杂度是多少？然后就知道面试官肯定不满意这个答案；② 两天前的日志怎么办？这个问题就真的没考虑过了。。。  5）程序引入的第三方库多了，就可能会有循环依赖，要求找出循环依赖的模块。  这个只知道是要找有向图中的环，从A出发能否再回到节点A，代码就写不出来了，然后就有些GG了。。。  2. 一面           一面问的很细致，按照简历上的内容一条条的过，包括项目和研究方向，我主要写了两个比赛，其中一个是研究生期间做的CCF一个对汽车评论文本进行基于视角的情感分析比赛，然后讲了是怎么做的，包括任务的切分：抽取汽车视角+判定情感分析。然后面试官问了用了什么方法来抽取视角，我主要是用了分词工具，然后问了是什么工具，接着问用怎么分析这个视角的情感分析，我当时主要是统计情感词，用字符串的正则化来处理，感觉没什么工作量，很水。。。          之后又问了欠拟合与过拟合，解释原因和解决办法，当时先答的欠拟合，主要原因是模型过于简单，不能捕捉到足够多的特征，解决方法在神经网络中可以是增加隐藏层等，在机器学习中，以SVM为例，当数据集线性不可分时，可以利用核函数，将数据映射到高维空间，变成线性可分。然后面试官问了SVM引入核函数的本质是啥，答曰从一个空间映射到另一个空间，面试官不满意，提示说其实是提高了维度，增加了模型复杂度（感觉自己总结能力有点弱）。          然后讲过拟合，主要原因是模型过于复杂，捕捉的特征太多（当时说的逻辑有些混乱==），然后解决方法有L2正则化，然后面试官顺势问了L2正则化的本质是什么，没答出来，面试官提醒说是限制解空间范围，缩小解空间，来控制模型复杂度（再次感觉自己基础也不扎实，总结的也不到位，这个其实之前有看到过，但是给忘了，或者没想到要这么答得）。面试官说这些在吴恩达的机器学习课程里都有（之前聊怎么学习的时候，有说过看公开课，看官方文档，感觉给自己挖了个坑），如果好好看了就都会知道的。感觉还是自己没有钻研下去吧，如果有静下心看的话，这些的确应该是知道的，能够想起来的。          之后因为自己简历比较寒碜，没有项目也没有实习经历，面试官就问了在学校都干了什么，自己有些虚，确实没什么工作量，平时也就是看看论文，找模型，试模型，结果模型也看的一知半解，就有些尴尬。面试官问最近情感分析都有什么方法，准确率大概是什么，用LSTM怎么对文本进行情感分析。这个前两个回答出来了，第三个当时看论文的时候就看了个大概，也没有深入学习LSTM，感觉回答的不好，自己需要注意下。          最后来个总结吧，感觉现场面试还是比电话面试要能看出更多东西的，也能发现自己在哪些方面做得不够好，比如我自己在基础知识上不够扎实，打算以后着重看神经网络相关的基础知识，然后再考虑实践检验。 |
| 美团成都一面总结  今天晚上面了美团成都，整体还是比较基础，属于基础面吧。总结一下(求赞。。。。)  1、 HTTP请求的过程，越清楚越好（我从dns解析到TCP三次握手，四次挥手，还有后台请求逻辑讲了一遍）  2、 Java线程池ThreadPoolEcecutor参数，基本参数，使用场景  3、 HashMap、HashTable、CurrentHashMap的区别及场景，说一下你的理解。HashMap的负载因子，干啥用的。  4、 ArrayList与LinkedList的区别及使用场景，有没有ArrayList增删效率比LinkedList高的场景。ArrayList底层采用什么实现数据的拷贝。  5、 MySQL的ACID讲一下，延伸到隔离级别  6、 JMM讲一下，volite关键字及使用场景。  7、 JVM的内存布局以及垃圾回收原理及过程讲一下。讲一下CMS垃圾收集器垃圾回收的流程。以及CMS的缺点（忘了。。）  8、 智力题（聊着聊着突然来个），两个杯子，一个有3升水，一个有5升水，请问如何倒出4升水。。（我回答说，使两者水平面一致。。。。一看到题，就这样啊。。但其答案不是我回答的这。。）  9、 手写二叉树的中序遍历与层序遍历  10、 手写二分查找（很基础）  11、最后还聊了一下项目，聊一下难点。  12、最后就是你有什么要问的。1、表现2、你们那边主要用那些java技术3、如果过，啥时通知下一面结果。  祝大家好运！整体还是比较基础的。。（58分钟） |
| 突如其来的阿里面试，25号刚进行的笔试，26号就进行了面试。。。。希望经验对大家有些帮助，楼主这种渣渣是注定与阿里无缘了。。。  一面：典型的基础面，一个小时左右的时间，感觉把java相关的东西都问了遍  上来本想自我介绍一下，结果不让自我介绍，说看过简历不要说了。。。  （1）实习经历介绍，实习中的项目的介绍，几乎每说一句都会被打断，会对其实现和原理进行提问（p.s：感觉不要乱说，说多了自己承受不住）  （2）JVM内存模型，JVM加载原理  （3）乐观锁，悲观锁  （4）IO实现及实现一下文件的复制  （5）NIO的原理  （6）多线程，线程与进程区别，结合内存说下  （7）如何实现数据的并发的安全  （8）讲下线程池的原理和简单实现  （9）设计模式  （10）java的网络通信的实现，简单实现个客户端服务端通信  （11）collection的理解，选择一种说下底层实现  （12）创建对象有哪些方式？为什么序列化？有什么好处  （13）同步与异步区别（这里楼主说了下JavaScript。。。结果就接下来进行了js的相关问题，后悔的不行）  （14）Hibernate的理解  （15）spring的理解，如何进行bean的配置，用ssh做没做过什么项目  （16）有什么想问我的  附上楼主博客，打波广告。。。。（<http://blog.csdn.net/megustas_jjc>）  二面。早上跟同学在买早饭，突如其来的面试电话，跟预想的一样，二面是项目面，感觉面试官真的很强，对底层原理问的很深。。。楼主表示招架不住，问题都是根据你简历上的项目或实习进行提问的：  （1）一句话概括这个框架的功能  （2）如何分配的线程数，怎么确保这是最优的方案  （3）java的自带线程池和你的这个有什么区别  （4）网络线程和业务线程怎么实现的通信  还有一些项目中的具体问题，主要是网络，协议和并发的一些相关问题，楼主在这方面的储备比较差。。。二面总共进行了14分钟。。。感觉已卒。。。牛客处女面试贴（总看大家的面试经历，是时候回报一波了）  总体感觉，阿里的面试官真的是很牛。。。技术问的很全面和深入。。。 |
|  |