第一题：谈谈Socket阻塞和非阻塞的理解

【回答】

socket是当内个，客户端向服务端发送请求，发送对象，序列化，用socket带到服务端。发挥response。socket会有两个缓冲区，write buffer，read buffer。

socket的定义，网络传输套接字，一个标志ID。ip:port

缓冲区有大小，读写操作会有争抢资源。有可能写满了，后面的被阻塞。如果写缓存写满了，后面的无法写入，非阻塞的话，个人感觉读缓存是非阻塞的。

【答案】

套接字对象实际上是放在操作系统内核中。这个套接字对象内部有两个重要的缓冲结构，一个是读缓冲(read buffer)，一个是写缓冲(write buffer)，它们都是有限大小的数组结构。当我们对客户端的socket写入字节数组，是将字节数组拷贝到内核区套接字对象的write buffer中，内核网络模块会有单独的线程负责不停地将write buffer的数据拷贝到网卡硬件，网卡硬件再将数据送到网线，经过一些列路由器交换机，最终送达服务器的网卡硬件中。同样，服务器内核的网络模块也会有单独的线程不停地将收到的数据拷贝到套接字的read buffer中等待用户层来读取。最终服务器的用户进程通过socket引用的read方法将read buffer中的数据拷贝到用户程序内存中进行反序列化成请求对象进行处理。

write buffer空间都是有限的，所以如果应用程序往套接字里写的太快，这个空间是会满的。一旦满了，写操作就会阻塞，直到这个空间有足够的位置腾出来。不过有了NIO(非阻塞IO)，写操作也可以不阻塞，能写多少是多少，通过返回值来确定到底写进去多少，那些没有写进去的内容用户程序会缓存起来，后续会继续重试写入。

同样read buffer的内容可能会是空的。这样套接字的读操作(一般是读一个定长的字节数组)也会阻塞，直到read buffer中有了足够的内容(填充满字节数组)才会返回。有了NIO，就可以有多少读多少，无须阻塞了。读不够的，后续会继续尝试读取。

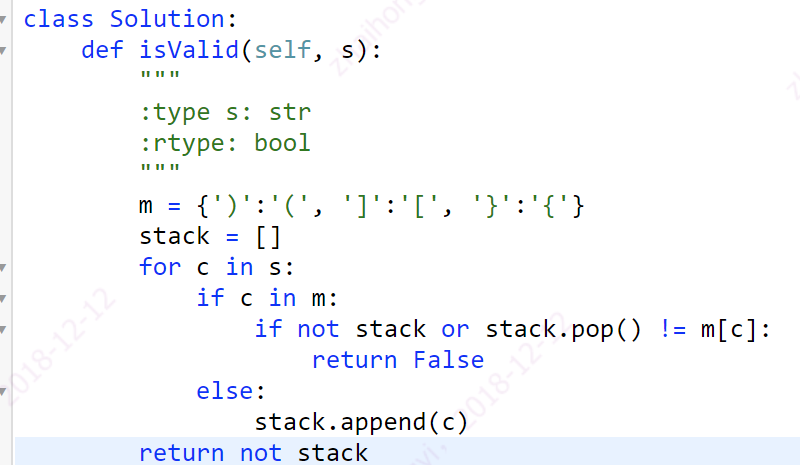
第二题：

给定一个只包括 '('，')'，'{'，'}'，'['，']' 的字符串，判断字符串是否有效。有效字符串需满足：

1. 左括号必须用相同类型的右括号闭合。
2. 左括号必须以正确的顺序闭合。

注意空字符串可被认为是有效字符串。

【答案】



第三题：谈谈你对swap的理解。swap的功能？Swapness的作用？Swap的优劣势？

【答案】见模拟面试20181205

【回答】嗯。。。swap嗯，让我稍微组织一下语言，拍，

当物理内存不足，操作系统会将不活跃的swap out出去。一部分是堆栈，匿名内存，另一部分是file，cache。可用内存 = free + cache，低于一个阈值会进行内存回收，回收两块，一个是匿名内存swapout，cache会执行刷盘或释放，是否有脏页。然后，swapness参数，配置两块内存回收的配比，越小，更少的使用swap，越高，主要使用swap，100的话1:1。优势：物理内存不足情况下弥补；劣势：刷磁盘，会有上下文切换，费时间。因为写磁盘所以耗时高。

第四题：谈谈你对重入锁，重入的理解

【答案】

对于同一个线程一把锁是可以多次获取的，获取多少次就要释放多少次。其他线程无法获取锁。synchronized关键字，即同步代码块也是可以重入的，如果递归调用同步代码块，是可以重复进入的。

【回答】

线程枪战的资源后，先获得锁，只有释放，其他线程才能使用资源。一个线程可以反复获得锁。释放多次。sycrnized。sychronized是可重入的。

什么时候会发生同步代码块的重入？当多个线程执行同步代码块。递归的时候。同步代码块不能进入递归。会锁死

职业规划：

我内个，互联网金融的业务开发，后面继续从事这方面的工作，熟门熟路，感兴趣。

兴趣点：生活当中接触的，交易列别、类型，不同的业务流程，有意思。