第一题：写一条shell停止Main.java进程

【答案】答案不唯一。jps -l | grep Main | awk ‘{print $1}’ | xargs kill

【回答】ps -ef|grep “main.java”|awk -F’’ {print $1}  exce `kill -9`

第二题：shell比较两个变量的大小

【答案】

x=45

y=45

if [ $x -gt $y ]

then

echo "x is bigger than y"

elif [ $y -gt $x ]

then

echo "y is bigger than x"

else

echo "x equal y"

fi

【回答】

A=1

B=3

if[ $A -gt $B]

echo "A 大于B";

else if[ $A -eq $B ]

echo "A 等于B";

else

echo "A 小于B";

fi

第三题：谈一谈延迟队列的工作原理

【答案】延迟队列使用优先级队列+重入锁实现。本质是延迟时间最小的优先级队列。入队对象需要实现getDelay和compareTo方法，分别返回剩余的延迟时间和比较延迟时间大小的方法。延迟队列提供了3组读写方法，分别是非阻塞、阻塞和带超时时间的阻塞方法。优先级队列默认使用了小顶堆，延迟时间最短的元素会在堆顶。当有元素出队或入队，堆通过siftDown和siftUp调整堆元素。堆是一颗完全二叉树，出队和入队时间复杂度O(logn)。

【回答】延迟队列，用到了重入锁和优先级队列来完成，能不能不要打内个了，，完成延迟队列元素到期才可以取出来。延迟队列元素要实现getdelay和compareto，放入油offer、put、add，offer首先获取锁，在优先级队列offer，优先级队列是小顶堆，放入后跟父节点比较，如果小于会交换位置，直到小于等于父节点。如果放入的元素，是队首，会leader，leader是thread，leader表示当前请求队首元素的线程，队首元素更新，leader置为null，通知请求竞争的线程。取元素，一种阻塞，一种非阻塞，take是非阻塞，没有返回null，也会获取释放锁。poll阻塞获取，实现逻辑是查看队首元素，不存在等待，存在判断等待时间，没过期继续等待，如果leader为空，成为leader，如果有leader，会继续等待，等待队首元素，等待通知竞争。poll之后如果取出来，之后还有队首元素，释放leader通知其他线程。等待之前会释放first，防止内存泄漏。时间复杂度？存O(logn)，取O(logn)。

第四题：谈一谈为什么需要分布式锁？分布式锁应该具有哪些特性？用redis如何实现？

【答案】见 分布式/分布式锁

【回答】

为什么需要分布式锁：分布式环境下的线程安全。抢购、秒杀，只有一个人能买到。备付金，一个备付金账户，同时只能有一个进程执行金额扣减。

特性：独占性，高可用，避免死锁，重入，阻塞或非阻塞，公平和非公平

redis实现分布式锁：kv的形式，存的是一个数值，map或set。多个进程需要抢占的资源。能get出来。。。用put方法，ex，当存在时put。用同一个key值，不存在可以存进去。有这个key说明已经锁了。用putNx。

增加过期时间，防止获取锁的进程挂了，防止其他进程无法获取锁

释放锁，put成功了，显示的清楚，clear

如何防止误释放锁，value里存一个信息，先判断value是不是我，是的话可以删除。

重入：第一次获得锁，标志已经获取锁，第二次用ex，put更改了value值，次数存在value中。

非公平的获取锁。

第五题：什么时候使用ThreadLocal？ThreadLocal是如何实现的？

【答案】见 Java/同步和并发/ThreadLocal

【回答】每个线程都想存一些信息，这些信息的属性名一样，同样的信息每个线程都有一个副本。打日志，一样的字段，每个线程的流水号和机构号不一样。会用到这个字段。

跟普通成员变量的区别？普通的成员变量在异常处理时，抛出异常的把成员变量传进去。

实现方法：不知道

线程私有，跟当前线程相关，每个线程拥有不一样的副本，KV的形式。map嵌套map。

第一个k跟线程，第二个k是想要的变量。感觉不用线程安全。