1、为什么使用ConcurrentHashMap？

（1）在多线程情况下，HashMap在执行put操作时会引起死循环，是因为多线程会导致HashMap的Entry链表形成环形数据结构，一旦形成环形数据结构，Entry的next节点永远不为空，就会产生死循环获取Entry。

（2）HashTable使用了synchronized来保证线程安全，但在线程竞争激烈的情况下，HashTable的效率会很低。

（3）ConcurrentHashMap使用了锁分段技术。将数据分成一段一段得存储，然后给每一段数据配一把锁，当一个线程占用锁访问其中一个段数据的时候，其他段的数据也能被其他线程访问。

2、ConcurrentHashMap 1.7是由Segment数组结构和HashEntry数组结构组成。Segment是一种可重入锁（ReentrantLock），在扮演锁的角色；HashEntry则用于存储键值对数据。一个Segment里包含一个HashEntry数组，每个HashEntry是一个链表结构的元素。

3、segments数组的长度ssize是通过concurrencyLevel计算得出。为了通过按位与的散列算法来定位segments数组的索引，必须保证segments数组的长度是2的N次方，假如concurrencyLevel等于14、15或16，ssize都会等于16，即容器里锁的个数也是16。

concurrencyLevel的最大值是65535，意味着segments数组的长度最大为65535，对应的二进制是16位。

4、