Zookeeper **安其内（quorum仲裁），攘其外。【CP原理 一致性和容错性】**

1角色 leader，follower , observer (观察者，需要在启动时进行配置，只进行查询和创建节点，不进行选举)

2 常用命令 zkServer.sh start / stop /status

Ls / set /delete /sync/ [watch 进行监控]

3 watcher 监视和通知 ，当节点信息变化时，能进行检测到变化，并通知客户端。（只运行一次回调函数）

4 创建节点大致流程

1. 客户端请求create操作并从follower 转发给leader
2. Leader进行征求意见【提议 proposal[prəˈpəʊzl]】
3. 反馈直到半数以上进行同意
4. Leader对该创建节点的操作进行commit操作，并更新事务id[zxid]
5. Flower 将返回结果返回给客户端

5 利用队列数据结构进行数据同步，每个flower 都有一个消息队列【操作事务】，flower消费消息队列里面的信息进行创建节点等操作来保持和主节点的一致性。

6 内部选主

1. 选主算法 先根据zxid进行选择，先选zxid大的，若zxid一致，再根据sid进行选择，sid大的获胜。

Zxid:【epoch(任期[iːpɒk])、counter】

1. 主节点和从节点都会进行心跳机制，当从节点（收、发）心跳时会重置时间闹钟，若在该规定时间内为收到心跳，则认为主节点失效，会重新进行选主。并更新新任选主的任期。
2. 假设老master恢复，会根据任期数进行判断，小于现有最大任期，则进行数据同步变为follower节点,根据uptodate进行差值化更新。

7 服务器为奇数的原因?

允许最大宕机的机器数在奇数节点和偶数节点是一致的，所以偶数个性价比不高。

8 zookeeper 实现分布式锁？

9 paxos -> raft -> zab ？ 仲裁算法