Paxos算法是由Leslie Lamport于1998年提出，其是一种基于消息传递的且具有高容错性的一致性算法。

1. Paxos产生背景。

在分布式系统中，我们总会遇到像是机器宕机、网络异常等问题。而paxos算法就是用来解决这类问题的（前提是保障数据的可靠性，信息不会被篡改），其可以快速且正确的对集群内部的某个数值（不只是狭义上的某个数。也可以是一条日志或是命令等等。）达成一致并保证系统的一致性。

1. Paxos相关概念。

Paxos算法共有三个角色，两个阶段。三个角色分别是：提议者（Proposer）:提出提案（Proposal）。其包括了提案编号（Proposal ID：全局递增的唯一id）和提议的值。决策者 (Acceptor，其个数大多数时候为奇数个)：整个算法中唯一的参与决策的角色，其回应Proposers的提案，当Proposers的提案被大多数的Acceptor接受时，则Proposal被批准了。最终决策学习者 (Learner):不参与决策，是充当记录员的一个角色，将上述两个角色最新达成一致的提案记录下来。注：再多副本的情况下，每个副本都可以具有这三种角色。两个阶段：prepare阶段：生成全局自增的proposal id，Proposer向所有Acceptors发送prepare请求（无需提案内容有id即可），Acceptors收到后将会对prepare请求做出promise承诺（1.不会接受小于等于proposal id的当前prepare的请求。2.不会接受小于proposal id的当前prepare的请求。）。Accept阶段：Proposer受到Acceptors的promise承诺后，其会向Acceptors发出Propose请求，Acceptors收到请求后会进行Accept处理。当Proposer收到Accept的请求之后就会将决议记录下来，并将其发送给所有的Learner。注：在不违背以前的承诺下，回复已经Accept过的提案中Proposal ID最大的那个提案的Value和Proposal ID，没有则返回空值。