# Paxos算法的理解

Paxos算法是一种在分布式系统中用于实现消息传输一致性的算法，这种算法包括三种角色分别为Acceptor（接收者），proposer（提议者），leaner（学习者）。现在举例说明paxos算法。

现有proposer：P1，P2；Accepter：A1，A2，A3。首先由P1和P2发出prepare请求，这个请求不携带value，这个请求的编号大于P1和P2各自已经提出编号的最大值，全局递增。比如现在A1，A2，A3已经响应过的最大Prepare请求分别为10，11，12。P1，P2发出的最大prepare请求编号不大于12，那么，P1，P2可以选择提案编号为13，14。在A1,A2,A3收到prepare请求后他们会判断这个提案的编号是否比自己已经响应过的最大请求编号大如果是，那么将新的提案编号设为最大提案编号，然后回复来自P1或者P2的prepare请求，回复时会带上各自原先响应过的最大提案的value，如果此时还没有接受过提案，那么返回空值，并且做出承诺，不会再接受比新的提案编号小的提案。在A1，A2，A3接受到prepare请求并响应后，P1，P2收到prepare回复，如果超过一半的acceptor回复的value为空值，比如A1，A2或者A1，A2，A3回复空值，那么发出accept请求，并且附带自己的value。

如果收到的回复数大于一半，但是有的不为空，那么将不为空的部分的value值返回给acceptor作为自己的value，对其余accptor则返回preposor自己的value值。比如收到A1的回复不为空值为value0，那么将这个value0作为自己的提议返回给accptor

如果收到的回复数小于一半那么回到准备阶段，重新发出prepare请求，重复以上步骤。

在下一个阶段，acceptor将回应P1的value，由于以上步骤产生的时间差可能会有新的P2发出的prepare请求被接受那么这个Accepter的最大响应编号会大于P1的提议编号，那么这个acceptor节点会对P1返回失败，否则返回成功。

在下一个阶段，preposer将统计成功的个数，如果成功个数大于一半，那么将他的value发送给所有的proposer和learner，否则产生更大的prepare请求编号，重复以上步骤，如果有的acceptor返回了失败，那么也将产生更大的prepare请求编号，重复以上所有步骤。