Paxos算法是一种在解决冲突时产生优先次序的算法。Paxos算法的特点是消息传递时高度容错即在运行中，对消息的可靠性不作要求，可以丢失、重复。

Paxos在系统中分别生成提案者 Proposer，接收者 Acceptor，和学习者 Learner三种。

Proposer的作用是提出提案 (Proposal)。Proposal信息包括提案编号 (PID) 和提案的值 (Value)。

Acceptor是参与决策的，对于Proposer提出的提案进行回复。收到Proposal后可以接受提案，若Proposal被多数的Acceptors选定，则称该Proposal被真实选定。

Learner：学习最新达成一致的提案。

Paxos算法通过一个决议分为两个阶段：

第一阶段：生成提案阶段，Proposer过程。Proposer向Acceptors发出请求，Acceptors对收到的请求进行接收。

1. Proposer向某个Acceptor集合发出请求，生成新的编号PID。在此情况下确保不再接受小于PID的提案，如果已接受过，响应已经接受于编号小于PID的最大编号提案。
2. 如果Proposer提出提案，半数以上的Acceptor响应，生成编号为PID，提案的值Value为V，[PID,V]，V为所有响应中编号最大的V；如果所有响应中无提案，Acceptor自行选择。当Acceptor处于接受过程时，可以忽略此时提出的任何请求。

第二阶段：接受提案阶段。

1. Proposer选择一个编号为PID，半数以上的Acceptor发送编号为PID的prepare的请求；或者一个Acceptor接收到一个编号为PID的prepare请求且PID大于这个Acceptor已经响应过的所有prepare请求编号，它就会将已经接受过的编号最大的提案作为响应反馈Propose不再接受编号小于PID的提案。
2. 如果Proposer接收到超过一半的Acceptor对其prepare响应，那么将[PID,V]中最大编号PID的提案值Value分发给半数以上的Acceptor；或者如果Acceptor接收到编号为PID的Acceptor请求，只要没有对编号为PID的Acceptor请求响应，则接受主Proposal。

第三阶段：决议分发阶段。Proposer在收到多数Acceptors的Accept之后，决议形成，将形成的决议发送给所有Learners。