Token Ring Algorithm:

Token Ring algorithm achieves mutual exclusion in a distributed system by creating a bus network of processes. A logical ring is constructed with these processes and each process is assigned a position in the ring. Each process knows who is next in line after itself. When the ring is initialized, process 0 is given a token. The token circulates around the ring. When a process acquires the token from its neighbor, it checks to see if it is attempting to enter a critical region. If so, the process enters the region, does all the work it needs to, and leaves the region. After it has exited, it passes the token to the next process in the ring. It is not allowed to enter the critical region again using the same token. If a process is handed the token by its neighbour and is not interested in entering a critical region, it just passes the token along to the next process.

Advantages:

- The correctness of this algorithm is evident.
- Only one process has the token at any instant, so only one process can be in a CS o Since the token circulates among processes in a well-defined order, starvation cannot occur.

Disadvantages

- Once a process decides it wants to enter a CS, at worst it will have to wait for every other process to enter and leave one critical region.
- If the token is ever lost, it must be regenerated. In fact, detecting that it is lost is difficult, since the amount of time between successive appearances of the token on the network is not a constant. The fact that the token has not been spotted for an hour does not mean that it has been lost; some process may still be using it.
- The algorithm also runs into trouble if a process crashes, but recovery is easier than in the other cases. If we require a process receiving the token to acknowledge receipt, a dead process will be detected when its neighbour tries to give it the token and fails. At that point the dead process can be removed from the group, and the token holder can pass the token to the next member down the line.

The token ring-based mutual exclusion algorithm is a well-known solution for coordinating access to shared resources in distributed systems. Its simple and efficient design ensures that only one process can access a shared resource at a time, thus preventing conflicts and ensuring consistency. While the algorithm can suffer from potential delays and network congestion, it remains a widely used and effective solution for achieving mutual exclusion in distributed systems.