

## Assignment No: 5 DS

Aim: Implement token based mutual Exclusion Algorithm.

Tools & environment: Java Runtime Environment, Java JDK.

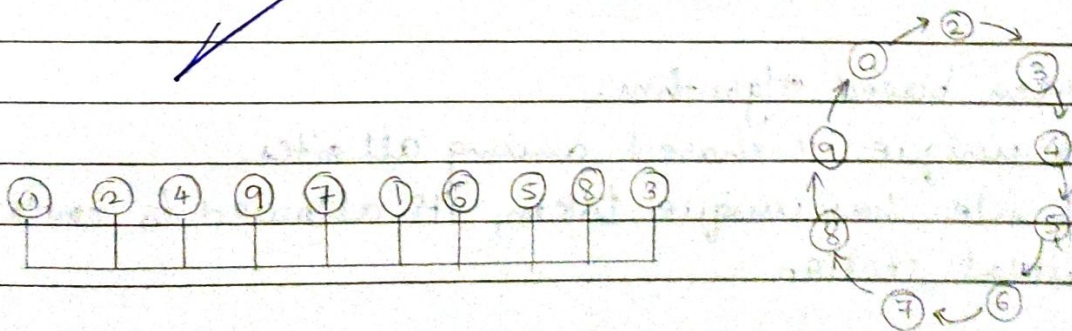
### Theory:

mutual exclusion is concurrency control property which is introduced to prevent race conditions. It is the requirement that process can't enter its critical section while another concurrent process is currently running.

Single computer system: memory & other resources are shared between different processes. The status is easily available in shared memory.

Eg: semaphore.

Distributed system: neither have shared memory nor physical clock. To eliminate this message passing is used.





## Requirements of mutual exclusion Algorithm:

No deadlock:

Two or more site should not endlessly wait for any message that will never arrive

No starvation:

Every site who wants to execute critical section should get an opportunity to execute critical section while other sites are executing critical section.

Each site should get a fair chance to execute critical section. Any request should be executed in the order they're made.

Fault Tolerance: Recognize failure without disruption

Solution to distributed mutual exclusion:

- As we know shared variables or local kernel can not be used to implement mutual exclusion.

I) Token based Algorithm:

- A unique token is shared among all sites. If site has unique token, it's allowed to enter critical section.

This approach uses seq no to order req for crit section.

Each request contains sequence no.



Each requests for critical section contains sequence no. This seq no is used to distinguish old & current request.

This approach insures mutual exclusion as token is unique.

Eg: Suzuki-Kasami's Broadcast Algorithm.

→ Conclusion: Hence Token Ring algorithm achieves mutual exclusion in distributed system by creating bus of network of processes.

