ASSIGNMENT NO. 05

TITLE: TOKEN RING

AIM: To implement token ring based mutual exclusion algorithm.

OBJECTIVE:
Students should be able to understand:-
- Basic concept of Mutual exclusion.
- Algorithm for Mutual exclusion.
- Token-Ring Algorithm.

PROBLEM STATEMENT:-

Write a program to implement token ring based mutual exclusion algorithm.

TOOLS / ENVIRONMENT:
- S/W:
  - Can be developed on any platform Windows /Linux.
  - C/C++/Java Language
- H/W:
  - Any basic configuration loaded machine (e.g. P IV)

THEORY:

Mutual exclusion

Fundamental to distributed systems is the concurrency and collaboration among multiple processes. In many cases, this also means that processes will need to simultaneously access the same resources. To prevent that such concurrent accesses corrupt the resource, or make it inconsistent, solutions are needed to grant mutual exclusive access by processes.

Overview

Distributed mutual exclusion algorithms can be classified into two different categories. In token-based solutions mutual exclusion is achieved by passing a special message between the processes, known as a token. There is only one token available and whoever has that token is allowed to access the shared resource. When finished, the token is passed on to a next process. If a process having the token is not interested in accessing the resource, it passes it on.

Token-based solutions have a few important properties. First, depending on how the processes are organized, they can fairly easily ensure that every process will get a chance at accessing the resource. In other words, they avoid starvation. Second, deadlocks by which several processes are indefinitely waiting for each other to proceed, can easily be avoided, contributing to their simplicity. The main drawback of token-based solutions is a rather serious one: when the token is lost (e.g., because the process holding it crashed), an intricate distributed procedure needs to be started to ensure that a new token is created, but above all, that it is also the only token. As an alternative, many distributed mutual exclusion algorithms follow a permission-based approach. In this case, a process wanting to access the resource first requires the permission from other processes. There are many

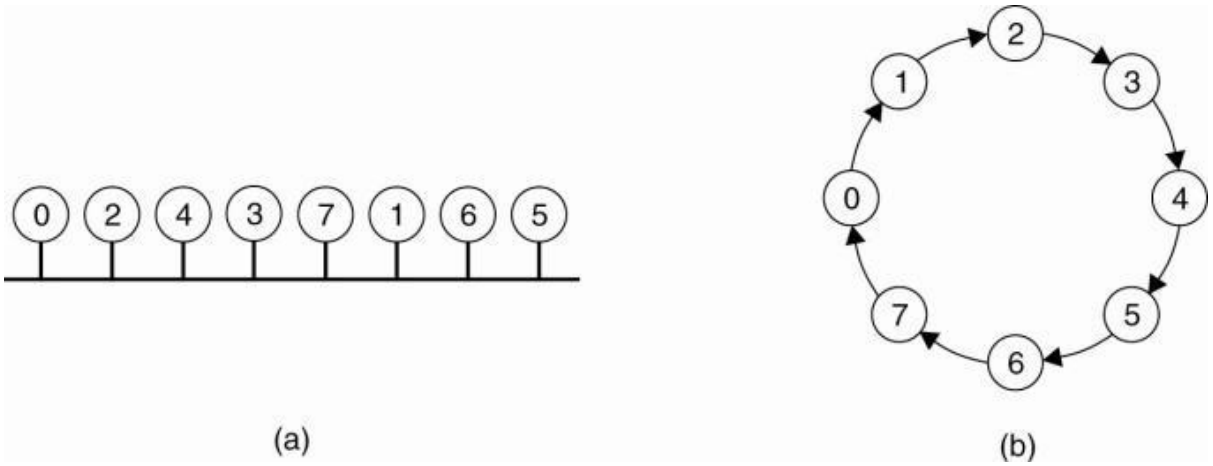
different ways toward granting such a permission and in the sections that follow we will consider a few of them.

Algorithm for Mutual exclusion:-

1. Centralized Algorithm
2. Distributed Algorithm
3. Token-Ring Algorithm
4. Decentralized Algorithm

Token-Ring Algorithm:-

Essence of Token-Ring Algorithm: Organize processes in a logical ring, and let a token be passed between them. The one that holds the token is allowed to enter the critical region (if it wants to)



(a) An unordered group of processes on a network. (b) A logical ring constructed in software.

- When the ring is initialized, process 0 is given a token.
- The token circulates around the ring.
- It is passed from process k to process k+1 (modulo the ring size) in point-to-point messages.
- When a process acquires the token from its neighbor, it checks to see if it needs to access the shared resource.
    - If so, the process goes ahead, does all the work it needs to, and releases the resources.
    - After it has finished, it passes the token along the ring.
    - It is not permitted to immediately enter the resource again using the same token.
- If a process is handed the token by its neighbor and is not interested in the resource, it just passes the token along.
    - As a consequence, when no processes need the resource, the token just circulates at high speed around the ring.

Advantages of Token-Ring Algorithm:

- Only one process has the token at any instant, so only one process can actually get to the resource.
- Since the token circulates among the processes in a well-defined order, starvation cannot occur.
- Once a process decides it wants to have access to the resource, at worst it will have to wait for every other process to use the resource.

Problems in Token-Ring Algorithm:

- If the token is ever lost, it must be regenerated
- Dead processes:
  - If a process receiving the token must acknowledge receipt, a dead process will be detected when its neighbor 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 throw the token over the head of the dead process to the next member down the line, or the one after that, if necessary.
  -

A Comparison of mutual exclusion algorithms:

Algorithm	Messages per entry/exit	Delay before entry (in message times)	Problems
Centralized	3	2	Coordinator crash
Decentralized	3mk, k = 1, 2,...	2 m	Starvation, low efficiency
Distributed	2(n - 1)	2(n - 1)	Crash of any process
Token ring	1 to	0 to n - 1	Lost token, process crash

Implementation:-

A fundamental algorithm in distributed systems consists of circulating a token amongst participants in a logical ring. Each participant executes as follows:

– wait receipt of token from its predecessor in the ring;
– enter <critical region>, if desired;
– send token to its successor in the ring.

The algorithm trivially satisfies two important properties:

(1) mutual exclusion is trivially guaranteed to the current holder of the token

(2) it allows fair access to the token by allowing each participant to access the token at most once in one traversal of the ring.

IMPLEMENTATION:

(Students should write here the implementation for their program. Students should attach printout of their programs with commands used to run the programs. Also attach the proper outputs of programs.)

CONLCUSION:

We successfully implemented Token-Ring Mutual Exclusion. This avoids Starvation. Lost Key is a major issue.

REFERENCES:

- "**Distributed Systems: Concepts and Design**" by George Coulouris, J Dollimore and Tim Kindberg, Pearson Education, ISBN: 9789332575226, 5th Edition, 2017
- **"Distributed Systems"**, Maarten van Steen, Andrew S. T, Third edition Version

FAQ:

1. Compare Mutual exclusion in single computer system V/S distributed system.
2. What are requirements of Mutual exclusion Algorithm?
3. Explain token-ring algorithm with suitable example?
4. What are different complications you face while implementing token-ring algorithm? Explain possible solutions for same.
5. Mention the criteria to evaluate the performance of algorithms for mutual exclusion