ASSIGNMENT NO. 7

Title: Election Algorithms

Aim: To Implement Bully and Ring algorithm for leader election.

Objective:

1. To enable distributed systems to select a leader in a decentralized manner, without requiring a centralized control mechanism.
2. To ensure that the leader selection process is reliable and efficient, even in the presence of failures, network delays, and other forms of uncertainty.
3. To provide a fair and deterministic method for selecting the leader, such that all nodes in the system have an equal chance of being chosen.

Tools / Environment:

Java Programming Environment, JDK 1.8, Eclipse Neon(EE).

Related Theory: Election Algorithm:

1. Many distributed algorithms require a process to act as a coordinator.
2. The coordinator can be any process that organizes actions of other processes.
3. *A* coordinator may fail.
4. How is a new coordinator chosen or elected?

Assumptions:

Each process has a unique number to distinguish them. Processes know each other's process number.\

There are two types of Distributed Algorithms:

1. Bully Algorithm
2. Ring Algorithm

Bully Algorithm:

A. When a process, P, notices that the coordinator is no longer responding to requests, it initiates an election.

1. P sends an ELECTION message to all processes with higher numbers.
2. If no one responds, P wins the election and becomes a coordinator.
3. If one of the higher-ups answers, it takes over. P"s job is done.

B. When a process gets an ELECTION message from one of its lower-numbered colleagues:

1. Receiver sends an OK message back to the sender to indicate that he is alive and will take over.

2. Eventually, all processes give up apart of one, and that one is the new coordinator.

3. The new coordinator announces *its* victory by sending all processes a **CO-ORDINATOR** message telling them that it is the new coordinator.

C. If a process that *was* previously down comes back:

1. It holds an election.
2. If it happens to be the highest process currently running, it will win the election and take over the coordinators job.

“Biggest guy" always wins and hence the name bully algorithm.

Ring Algorithm

Initiation:

1. When a process notices that coordinator is not functioning:

2. Another process (initiator) initiates the election by sending "ELECTION" message (containing its own process number)

Leader Election:

3. Initiator sends the message to it's successor (if successor is down, sender skips over it and goes to the next member along the ring, or the one after that, until a running process is located).

4. At each step, sender adds its own process number to the list in the message.

5. When the message gets back to the process that started it all: Message comes back to initiator. In the queue the **process with maximum ID Number wins**.

Initiator announces the winner by sending another message around the ring.

Designing the solution: A. For Ring Algorithm

Initiation:

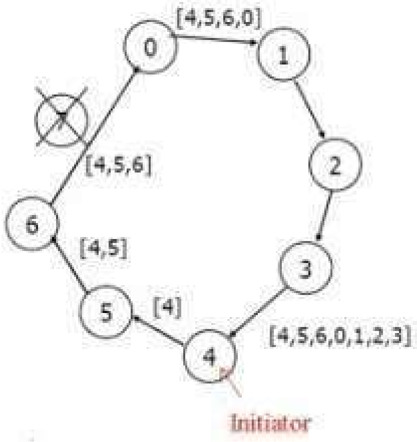
1. Consider the Process 4 understands that Process 7 is not responding.

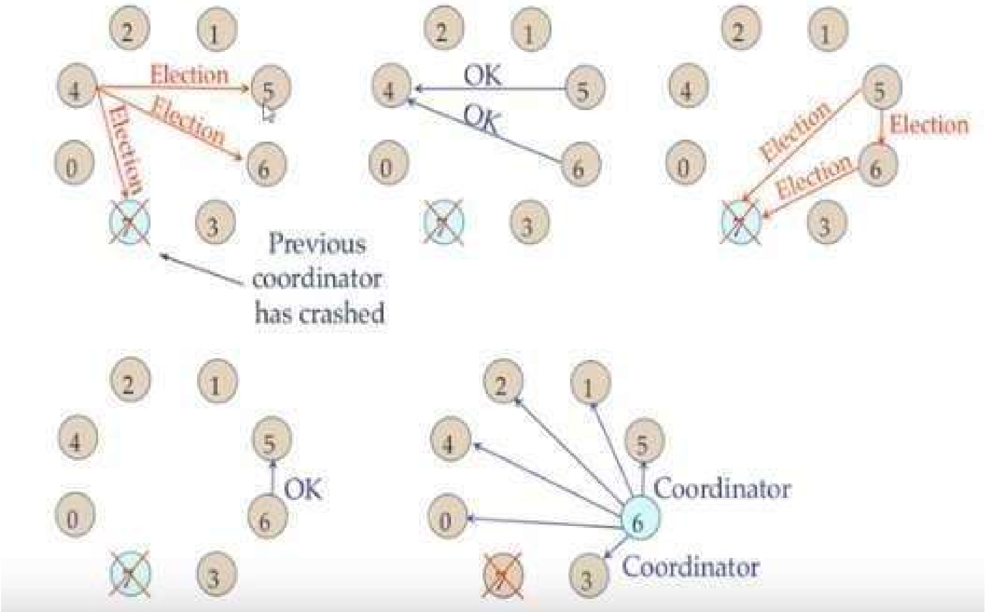
2.Process 4 initiates the Election by sending "ELECTION" message to it's successor (or next alive process) with it's ID.

Leader Election:

3.Messages comes back to initiator. Here the initiator is 4.

4. Initiator announces the winner by sending another message around the ring. Here the process with highest process ID is 6. The initiator will announce that Process 6 is Coordinator.





Implementing the solution:

For Ring Algorithm:

1. Creating Class for Process which includes
    i) State: Active / Inactive

    ii) Index: Stores index of process.
    iii) ID: Process ID

2. Import Scanner Class for getting input from Console
3. Getting input from User for number of Processes and store them into object of classes.
4. Sort these objects on the basis of process id.
5. Make the last process id as "inactive".
6. Ask for menu 1.Election 2.Exit
7. Ask for initializing election process.

8.These inputs will be used by Ring Algorithm.

Conclusion:

Election algorithms **are designed to choose a coordinator.** We have two election algorithms for two different configurations of distributed system. **The Bully** algorithm applies to system where every process can send a message to every other process in the system and **The Ring** algorithm applies to systems organized as a ring (logically or physically). In this algorithm we assume that the link between the process are unidirectional and every process can message to the process on its right only.

Outcome:

1. Students learned the fundamentals of process coordinator election algorithms in DS
2. Students developed Bully and Ring algorithm for leader election

FAQ:

1. Who is process coordinator? What are its responsibilities?
2. Need of Election Algorithm?
3. What is centralized and decentralized algorithm?
4. Explain Election working of algorithm for Ring & Bully?
5. What is „Token"?
6. Why algorithm is known as "Bully"?