Roll No: 31443 DOP: 11 Oct, 2021

Assignment No : 3 (DS)

1) Title: Simulation of Election Algorithm (Ring and Bully Algorithm)

2) Software and hardware requirement:

* Software requirements
1) JDK / Java Development kit
2) Code Editor

* Hardware requirements
1) Computer System:

Processor : is 7th Gen.
Ram : 8GB

2) I/O Peripherals : Keyboard and mouse
3) Monitor : 720p / 1080p.

3) Learning Objective:

1) To understand the election Algorithm; Bully & Ring Algorithm.

2) Understand the implementational perspective using suitable language and datastructure

4) Learning Outcome

1) One will be able to define the election algorithm.

2) One will be able to implement the Ring / Bully Algorithm.

5) Concept related Theory:

Distributed Algorithm is a algorithm that runs on a distributed systems. Distributed system is a collection of independent computers that do not share their memory. Each processor has its own memory and they communicate via communication networks.

Communication in networks is implemented in a process on one machine communicating with a process on other machine.

Many Algorithm used in distributed system requires a co-ordinator that performs function needed by other processes in the system.

Election Algorithms are used or designed to choose a co-ordinator.

Election Algorithm:

Election algorithm choose a process from group of processors to act as a co-ordinator. If the co-ordinator process crashes due to some reasons, then a new co-ordinator is elected on other processor. Election algorithm basically determines where a new copy of co-ordinator should be restarted.

Election algorithm assumes that every active process in the system has a unique priority number. The process with highest priority will be chosen as a new co-ordinator. Hence, when a co-ordinator fails, this algorithm elects that active process which has highest priority number. Then this number is send to every active process in the distributed system.

6) We have two election algorithm for two different configuration of distributed systems.

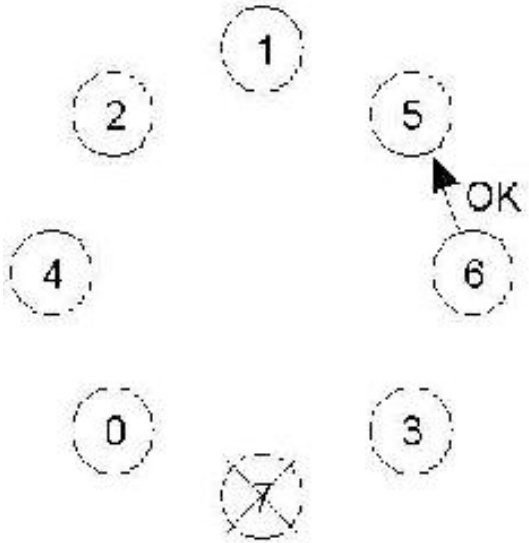
1) The Bully Algorithm

This algorithm applies to system where every process can send a message to every other process in the system.

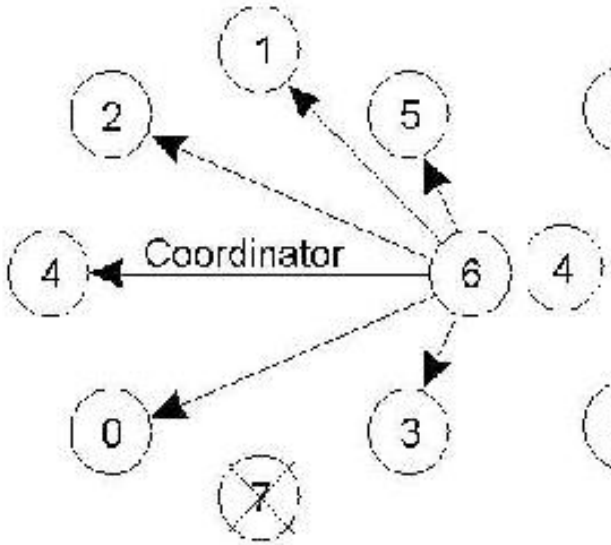
Algorithm: Suppose process P sends a message to co-ordinator

1) If co-ordinator does not respond to it within time interval T, then it is assumed that coordinator has crashed/failed.
2) Now process P sends election message to every process with high priority number.
3) It waits for responses, if no one responds for time interval T then process P elects itself as a coordinator.
4) Then it sends a message to all lower priority number ~~processes~~ processes that it is elected as their new co-ordinator.
5) However, if an answer is received within time T from any other process Q.
   a) Process P again waits for time interval T to receive another message from Q that it has been elected as co-ordinator.
   b) If Q doesn't responds with time interval T then it is assumed to have failed and algorithm is then restarted.

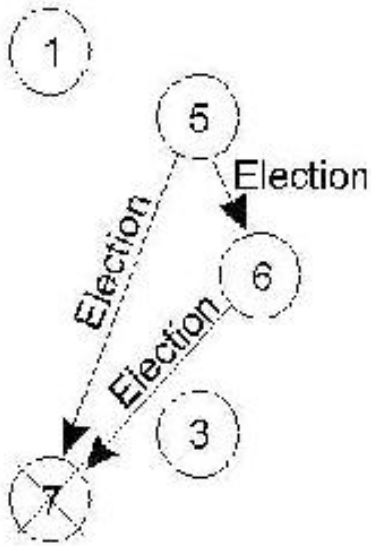
This algorithm is a method for dynamically electing a co-ordinator or leader from a group of distributed computer processes.


1
2
5
OK
4
6
0
3
7

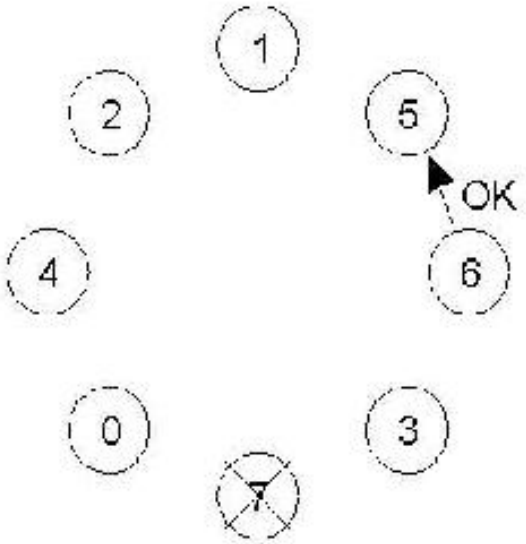
(d)


1
2
5
4
Coordinator
6
0
3
7

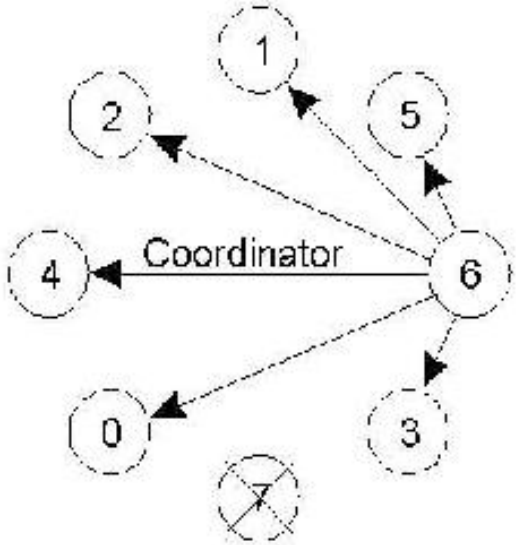
(e)


1
5
Election
Election
6
Election
3
7

(c)


1
2
5
OK
4
6
0
3
7

(d)


1
2
5
4
Coordinator
6
0
3
7

(e)

2) Ring Algorithm.

This algorithm applies to systems organized as a ring (logically or physically). In this algorithm we assume that the link between the process are unidentified unidirectional and every process can message to the process on its right only.

Data structure that this algorithm uses is active list, a list that has priority number of all active processes in the system.

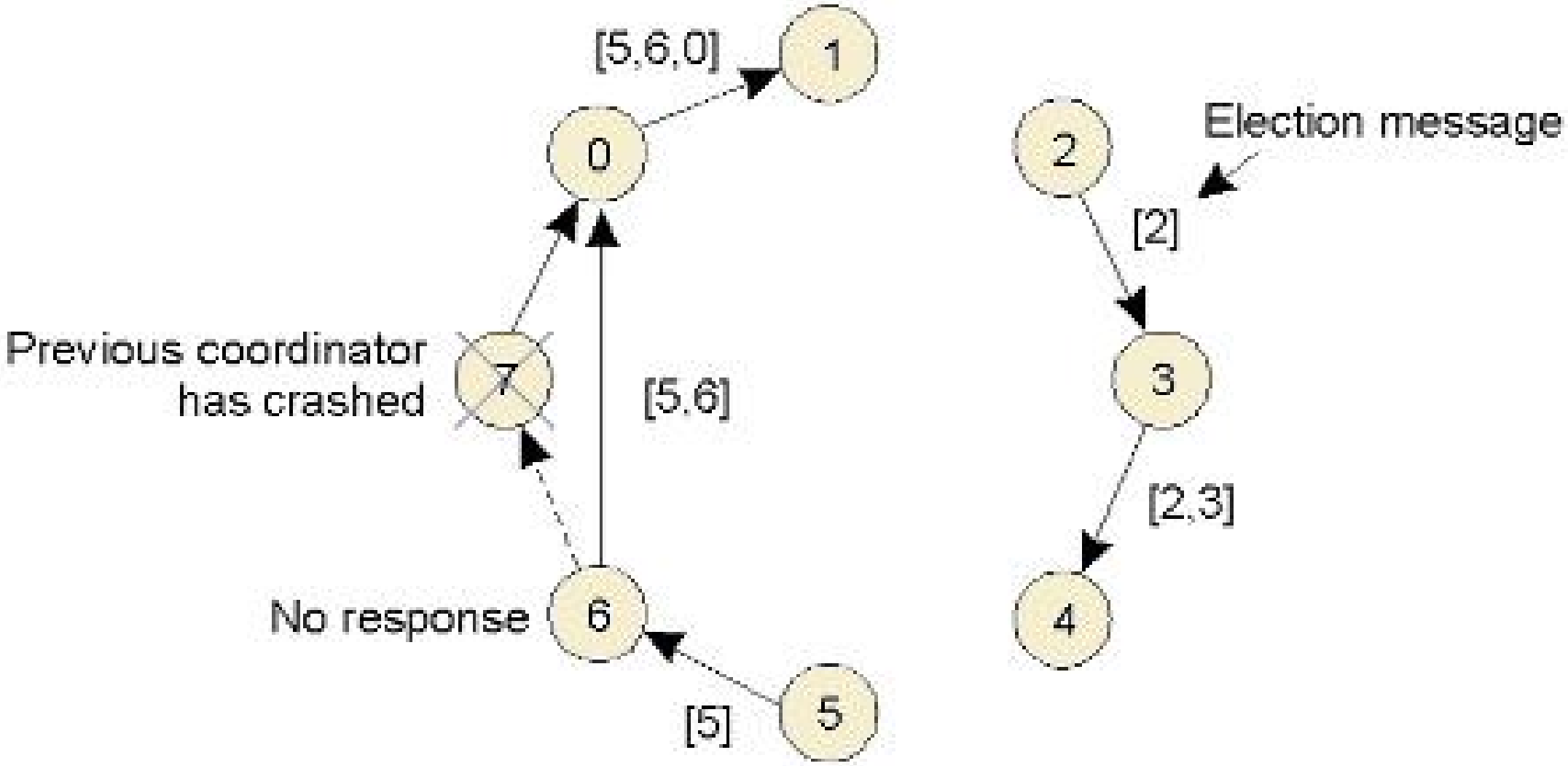
Algorithm:

1) If process P1 detects a co-ordinator failure, it creates new active list which is empty initially. it sends election message to its neighbor on right and adds number 1 to its active ~~list~~ list

2) If process P2 receives message elect from processes on left, it responds in 3 ways:

(i): If message received does not contain 1 in active list and forword message

(ii) If this is the first election message it has received or sent, P1 creates new active list with numbers 1 & 2. it then sends election message 1 followed by 2.


[5,6,0]
1
0
Election message
2
[2]
Previous coordinator
has crashed
7
[5,6]
3
[2,3]
No response
6
4
[5]
5

(iii) : If process P1 receives its own election message 1 then active list for P1 now contains number of all the active processes in the system. Now process P1 detects highest priority number from list elects it as the new co-ordinator.

7) Conclusion:

Understood the concept of Election Algorithm, and was ~~able~~ able to implement it using appropriate data structure and language.

8) References:

1) Geeks for Geeks
2) Youtube.