

Experiment No: 05

Aim: Write a program to implement Bully Election Algorithm.

Theory:

The bully algorithm is a method in distributed computing for dynamically selecting a coordinator by process ID number.

When a process P determines that the current coordinator is down because of message timeouts or failure of the coordinator to initiate a handshake, it performs the following sequence of actions:

1. P broadcasts an election message (inquiry) to all other processes with higher process IDs.
2. If P hears from no process with a higher process ID than it, it wins the election and broadcasts victory.
3. If P hears from a process with a higher ID, P waits a certain amount of time for that process to broadcast itself as the leader. If it does not receive this message in time, it re-broadcasts the election message.

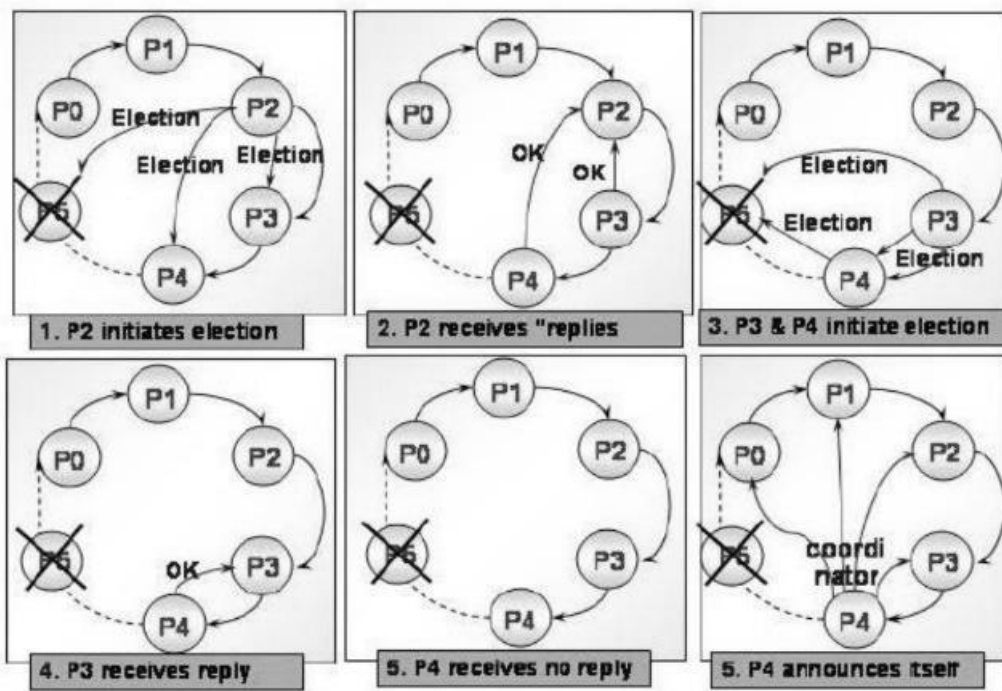
Note that if P receives a victory message from a process with a lower ID number, it immediately initiates a new election. This is how the algorithm gets its name - a process with a higher ID number will bully a lower ID process out of the coordinator position as soon as it comes online.

The bully algorithm is a method in distributed computing for dynamically selecting coordinator by process ID number. The Bully Algorithm was devised by Garcia-Molina in 1982.

The following diagram shows the working of Bully Algorithm: -

Performance of Bully Algorithm

- Best case scenario: The process with the second highest id notices the failure of the coordinator and elects itself.
 - N-2 coordinator messages are sent.
 - Turnaround time is one message transmission time.
- Worst case scenario: When the process with the least id detects the failure.
 - N-1 processes altogether begin elections, each sending messages to processes with higher ids.
 - The message overhead is $O(N^2)$.
 - Turnaround time is approximately 5 message transmission times.



Algorithm:

When a process notices that the coordinator is no longer responding to requests, it initiates an election. Process P, holds an election as follows:

1. P sends an ELECTION message to all processes with higher numbers.
2. If no one responds, P wins the election and becomes coordinator.
3. If one of the higher-up's answers, it takes over. P's job is done.
4. At any moment, a process can get an ELECTION message from one of its lower numbered colleagues.
5. When such a message arrives, the receiver sends an OK message back to the sender to indicate that it is alive and will take over.
6. The receiver then holds an election, unless it is already holding one.
7. Eventually, all processes give up but one, and that one is the new coordinator.

Code:

Bully.java

```

1  import java.util.Scanner;
2  public class BullyAlgorithm{
3      static int n;
4      static int[] pro = new int[100];
5      static int[] sta = new int[100];
6      static int co;
7      public static void main(String[] args) {
8          System.out.print("Enter number of processes: ");
9          Scanner sc = new Scanner(System.in);
10         n = sc.nextInt();
11         int i, j, c, cl = 1;
12         for(i = 0; i < n; i++){
13             sta[i] = 1;
14             pro[i] = i;
15         }
16         boolean choice = true;
17         int ch;
18         do{
19             System.out.println("Enter your choice: ");
20             System.out.println("1. crash process");
21             System.out.println("2. recover process");
22             System.out.println("3. exit");
23             ch = sc.nextInt();
24             switch (ch) {
25                 case 1:
26                     System.out.print("Enter the process number ");
27                     c = sc.nextInt();
28                     sta[c-1] = 0;
29                     cl = 1;
30                     break;
31                 case 2:
32                     System.out.print("Enter the process number ");
33                     c = sc.nextInt();
34                     sta[c-1] = 1;
35                     cl = 1;
36                     break;
37                 case 3:
38                     choice = false;
39                     cl = 0;
40                     break;
41             }
42             if(cl == 1){
43                 System.out.print("Which process will initiate the election? = ");
44                 int ele = sc.nextInt();
45                 elect(ele);
46             }
47             System.out.println("Filnal co-ordinator is "+co);
48         }while(choice);
49     }
50     static void elect(int ele){
51         ele = ele - 1;
52         co = ele+1;
53         for(int i = 0; i < n; i++){
54             if(pro[ele] < pro[i]){
55                 System.out.println("Election message is sent from "+ (ele+1)+ " to "+ (i+1));
56                 if(sta[i] == 1){
57                     System.out.println("Ok message is sent from "+(i+1)+" to "+(ele+1));
58                 }
59                 if(sta[i] == 1){
60                     elect(i+1);
61                 }
62             }
63         }
64     }
65 }

```

Output:

```
D:\Practical>java BullyAlgorithm
Enter number of processes: 5
Enter your choice:
1. crash process
2. recover process
3. exit
1
Enter the process number 1
Which process will initiate the election? = 2
Election message is sent from 2 to 3
Ok message is sent from 3 to 2
Election message is sent from 3 to 4
Ok message is sent from 4 to 3
Election message is sent from 4 to 5
Ok message is sent from 5 to 4
Election message is sent from 3 to 5
Ok message is sent from 5 to 3
Election message is sent from 2 to 4
Ok message is sent from 4 to 2
Election message is sent from 4 to 5
Ok message is sent from 5 to 4
Election message is sent from 2 to 5
Ok message is sent from 5 to 2
Final co-ordinator is 5
Enter your choice:
1. crash process
2. recover process
3. exit
1
```

```
2
Enter the process number 1
Which process will initiate the election? = 3
Election message is sent from 3 to 4
Ok message is sent from 4 to 3
Election message is sent from 4 to 5
Election message is sent from 3 to 5
Final co-ordinator is 4
Enter your choice:
1. crash process
2. recover process
3. exit
3
Final co-ordinator is 4

D:\Practical>
```

Conclusion: We have successfully studied about Election algorithm and implemented Bully election algorithm in Distributed System.