INSTITUTE OF ENGINEERING

ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT

Kupondole, Lalitpur

(AFFILIATED TO TRIBHUVAN UNIVERSITY)

Lab No:4

Subject: Distributed System

Submitted By:

Name: Sameep Dhakal

Roll no: ACE074BCT063

Date: 16/07/2021

Submitted To:

Department of Computer
and
Electronics Engineering

Lab:4

1. Title

Application of the bully algorithm

2. Objective

The objective of this lab is implement the Bully algorithm

3. Software Used

The Java Development Kit (JDK) was used as a compiler, notepad was used to write the code and the windows command prompt was used to execute the code.

4. Introduction

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

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 the coordinator.

If coordinator does not respond to it within a time interval T, then it is assumed that coordinator has failed.
Now process P sends election message to every process with high priority number.
It waits for responses, if no one responds for time interval T then process P elects itself as a coordinator.
Then it sends a message to all lower priority number processes that it is elected as their new coordinator.
However, if an answer is received within time T from any other process Q,

(I) Process P again waits for time interval T’ to receive another message from Q that it has been elected as coordinator.
(II) If Q doesn’t respond within time interval T’ then it is assumed to have failed and algorithm is restarted.

5. Code Implementation

```
import java.io.*;
import java.util.Scanner;
class Bully{
  static int n;
  static int pro[] = new int[100];
  static int sta[] = new int[100];
  static int co;

  public static void main(String args[])throws IOException{
        System.out.println("Enter the number of process");
        Scanner in = new Scanner(System.in);
        n = in.nextInt();

        int i,j,k,l,m;

        for(i=0;i<n;i++)
        {
              System.out.println("For process "+(i+1)+":");
              System.out.println("Status:");
              sta[i]=in.nextInt();
              System.out.println("Priority");
              pro[i] = in.nextInt();
        }
        System.out.println("Which process will initiate election?");
```

```
        int ele = in.nextInt();
        elect(ele);
        System.out.println("Final coordinator is "+co);
        }

        static void elect(int ele){
                ele = ele-1;
                co = ele+1;
                for(int i=0;i<n;i++)
                {
                        if(pro[ele]<pro[i])
                        {
                                System.out.println("Election message is sent from "+(ele+1)+"
to "+(i+1));
                                if(sta[i]==1)
                                        elect(i+1);
                        }
                }
        }
 }
```

6. Output

```
C:\Users\sameep\Desktop>java Bully.Java
Enter the number of process
7
For process 1:
Status:
1
Priority
1
For process 2:
Status:
1
Priority
2
For process 3:
Status:
1
Priority
3
For process 4:
Status:
0
Priority
4
For process 5:
Status:
0
Priority
5
For process 6:
Status:
1
Priority
6
For process 7:
Status:
1
Priority
7
Which process will initiate election?
3
Election message is sent from 3 to 4
Election message is sent from 3 to 5
Election message is sent from 3 to 6
Election message is sent from 6 to 7
Election message is sent from 3 to 7
Final coordinator is 7

C:\Users\sameep\Desktop>
```

6. Discussion

We implemented the bully algorithm in java. We first compiled the java file and then we run it to analyze the output in command prompt. We placed status and priority to different processes available and finally got our coordinator process.

7. Conclusion

Hence, in this lab we learned about the bully algorithm and its implementation.