Page No.
Date

Aim- Implementation of Bully Election algorithm

Theory:
Election algorithm are designed to choose a co-ordinator. Any process can serve as a co-ordinator. Any process can call an election

Bully algorithm:-
In a distributed computing, the bully algorithm is a method for dynamically electing a co-ordinator or leader from a group of distributed computer process.
The process with the highest process ID number from amongst the non failed process is elected as the co-ordinator

The algorithm uses the following message types:-

① Election message :- Sent to announce election

② Answer message - Responds to the election message

③ Co-ordinator message - Sent by the winner of the election to announce victory

Teacher's Sign.: _____

Page No.
Date

when a process p recovers from failure or the failure detector indicates that the current co-ordinator has failed, P perform the following actions.

1) If P has the highest process id, it sends a victory message to all other process & becomes the new co-ordinator, otherwise P broadcasts an Election message to all other process with higher process IDs than itself.

2) If P receives no answer after sending an election message, then it broadcasts a victory message to all process & becomes the co-ordinator.

3) If P receives answer with higher ID process it sends no further message for this election & waits for a victory message.
If no victory message, If there is no victory message after a period of time, it restarts the process.

4) If P receives a co-ordinator message, it treats the sender as co-ordinator.

Conclusion - we studied & implemented Bully election algorithm.

Teacher's Sign.: _____

```
def election():
   ini = int(input("Enter the Initiator\t"))
   for j in range(ini,n):
      print("Process p",j," called for election")
   for k in range(ini,n):
      if(status[k]==1):
         print("Process p",k," is In")
      else:
         print("Process p",k," is Dead")
   print("*** New Coordinator is Process p",n-2,"
***")


item=0
n = int(input("Enter no of Processes:\t"))
coordinator = n-1
status=[]
for i in range(n):
   status.append(1)
while(item<4):

item=int(input("Choose\n1.Crash\n2.Recover\n3.
Display Coordinator\n4.Exit\t"))
   if(item==1):
      crash = int(input("Enter the Process
number u want to crash:\t"))
      status[crash]=0
      if(crash==n-1):
         print("Coordinator has crashed")
         election()
         coordinator = n-2
   elif(item==2):
      recover = int(input("Enter Process which is
Recovered\t"))
      status[recover]=1
      if(recover == n-1):
         coordinator=n-1
         print("*** New Coordinator is Process
p",coordinator," ***")
   elif(item==3):
      print("*** Coordinator is Process
p",coordinator," ***")
   else:
      print("BYE BYE")
      break
```

```
D:\D17B-6,8\BullyElection>python bully_election.py
Enter no of Processes:  7
Choose
1.Crash
2.Recover
3.Display Coordinator
4.Exit  1
Enter the Process number u want to crash:       6
Coordinator has crashed
Enter the Initiator     4
Process p 4  called for election
Process p 5  called for election
Process p 6  called for election
Process p 4  is In
Process p 5  is In
Process p 6  is Dead
*** New Coordinator is Process p 5  ***
Choose
1.Crash
2.Recover
3.Display Coordinator
4.Exit  3
*** Coordinator is Process p 5  ***
Choose
1.Crash
2.Recover
3.Display Coordinator
4.Exit  2
Enter Process which is Recovered        6


*** New Coordinator is Process p 6  ***
Choose
1.Crash
2.Recover
3.Display Coordinator
4.Exit  1
Enter the Process number u want to crash:       5
Choose
1.Crash
2.Recover
3.Display Coordinator
4.Exit  2
Enter Process which is Recovered        5
Choose
1.Crash
2.Recover
3.Display Coordinator
4.Exit  3
*** Coordinator is Process p 6  ***
Choose
1.Crash
2.Recover
3.Display Coordinator
4.Exit  4
BYE BYE
```