***Name: …………………………………***

***Department. :*** *Computer Engineering*

***Class & Semester:*** *B.E (Final Year), SEM VIII*

***Subject:*** *Distributed Computing Lab (DCL)*

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***Expt. No. 06***

***Title:*** *Write a program to demonstrate Bully Election Algorithm.*

***Date:***

***Subject In-charge Sign:***

***…………………………….***

**Experiment No. 06**

**Aim:** Write a program to Demonstrate Bully Election Algorithm.

**Theory:**

The Bully Algorithm was devised by Garcia-Molina in 1982. 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-ups answers, it takes over. P's job is done.

At any moment, a process can get an ELECTION message from one of its lower-numbered colleagues. 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. The receiver then holds an election, unless it is already holding one. Eventually, all processes give up but one and that one is the new coordinator. It announces its victory by sending all processes a message telling them that starting immediately it is the new coordinator.

If a process that was previously down comes back up, it holds an election. If it happens to be the highest-numbered process currently running, it will win the election and will take over the coordinator's job. Thus the biggest guy in town always wins, hence the name "Bully Algorithm".

**Program:**

**import** java**.**io**.\*;**

class BullyAlgo

**{**

int cood**,**ch**,**crash**;**

int prc**[];**

public void election**(**int n**)** **throws** IOException

**{**

BufferedReader br**=new** BufferedReader**(new** InputStreamReader**(**System**.**in**));**

System**.**out**.**println**(**"\n The COrdinator Has Crashed"**);**

int flag**=**1**;**

**while** **(**flag**==**1**)**

**{**

crash**=**0**;**

**for(**int i1**=**0**;**i1**<**n**;**i1**++)**

**if(**prc**[**i1**]==**0**)**

crash**++;**

**if(**crash**==**n**)**

**{**

System**.**out**.**println**(**"\n\*\*\*All processes are crashed \*\*"**);**

**break;**

**}**

**else**

**{**

System**.**out**.**println**(**"\nEnter the Intiator"**);**

int init**=**Integer**.**parseInt**(**br**.**readLine**());**

**if((**init**<**1**)||(**init**>**n**)||(**prc**[**init**-**1**]==**0**))**

**{**

System**.**out**.**println**(**"\nInvalid Intiator"**);**

**continue;**

**}**

**for(**int i1**=**init**-**1**;**i1**<**n**;**i1**++)**

System**.**out**.**println**(**"Process"**+(**i1**+**1**)+**" called for election"**);**

System.out.println("");

for(int i1=init-1;i1<n;i1++)

{

if(prc[i1]==0)

System.out.println("Process"+(i1+1)+"Is Dead");

else

System.out.println("Process"+(i1+1)+"Is in");

}

for(int i1=n-1;i1>=0;i1--)

if(prc[i1]==1)

{

cood=(i1+1);

System.out.println("\n\*\*\* New Coordintaors Is "+(cood)+"\*\*\*\*");

flag=0;

break;

}

}

}

}

public void Bully( ) throws IOException

{

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the number of prcoess:");

int n=Integer.parseInt(br.readLine());

prc=new int[n];

crash=0;

for(int i=0;i<n;i++)

prc[i]=1;

cood=n;

do

{

System.out.println("\n\t 1.Crash a process");

System.out.println("\n\t 2.Recover a process");

System.out.println("\n\t 3.Display New Coordintor");

System.out.println("\n\t 4.Exit");

ch=Integer.parseInt(br.readLine());

switch(ch)

{

case 1:System.out.println("Enter a process to crash ");

int cp=Integer.parseInt(br.readLine());

if((cp>n)||(cp<1))

{

System.out.println("Invalid Process!Enter a valid process");

}

else if((prc[cp-1]==1)&&(cood!=cp))

{

prc[cp-1]=0;

System.out.println("\nProcess"+cp+"Has been Crashed");

}

else if((prc[cp-1]==1)&&(cood==cp))

{

prc[cp-1]=0;

election(n);

}

else

System.out.println("\nProcess"+cp+"Is already Crashed");

break;

case 2: System.out.println("\n Crashed Processes Are: \n");

for(int i=0;i<n;i++)

{

if(prc[i]==0)

System.out.println(i+1);

crash++;

}

System.out.println("Enter The Process You Want To Recover");

int rp=Integer.parseInt(br.readLine());

if((rp<1)||(rp>n))

System.out.println("\n Invalid Process. Enter A Valid ID");

else if((prc[rp-1]==0)&&(rp>cood))

{

prc[rp-1]=1;

System.out.println("\n Process "+rp+" Has Recovered");

cood=rp;

System.out.println("\n Process "+rp+" Is The New Coordinator ");

}

else if(crash==n)

{

prc[rp-1]=1;

cood=rp;

System.out.println("\n Process "+rp+" Is The New Coordinator ");

crash--;

}

else if((prc[rp-1]==0)&&(rp<cood))

{

prc[rp-1]=1;

System.out.println("\n Process "+rp+" Has Recovered");

}

else

System.out.println("\n Process "+rp+" Is Not A Crashed Process");

break;

case 3:System.out.println("\nCurrent Coordinator Is "+cood);

break;

case 4:System.exit(0);

break;

default:System.out.println("\n Invalid Entry!");

break;

}

}

while(ch!=4);

}

public static void main(String args[]) throws IOException

{

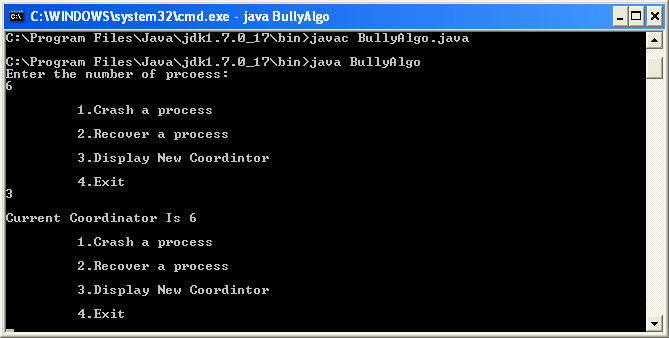
BullyAlgo ob=new BullyAlgo();

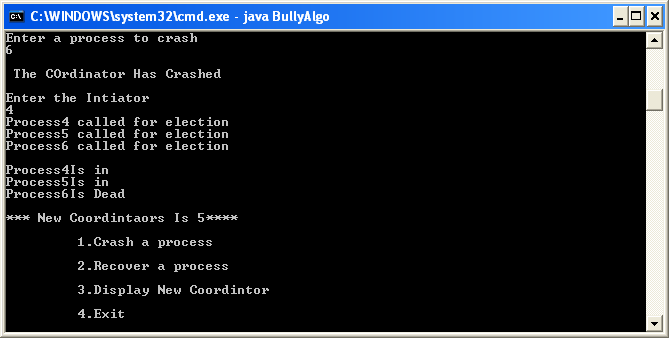
ob.Bully();

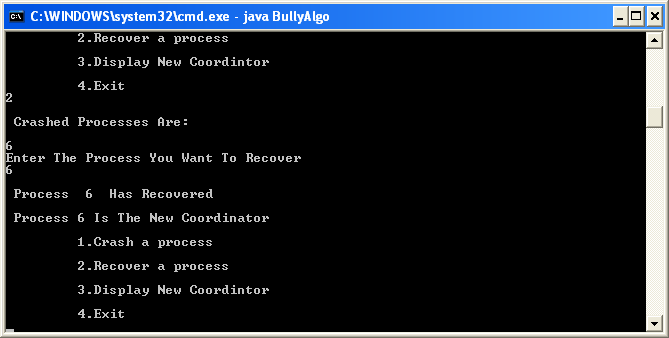
}

}

**Output:**







**Conclusion:** Thus we have studied a program to demonstrate Bully Election Algorithm.