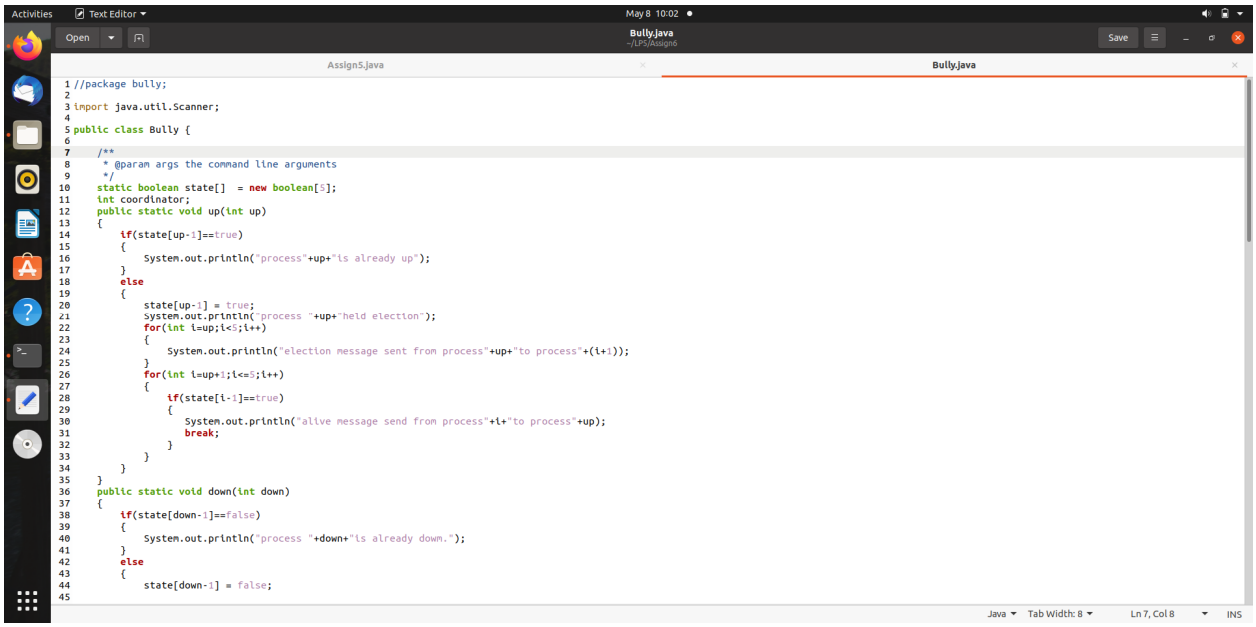
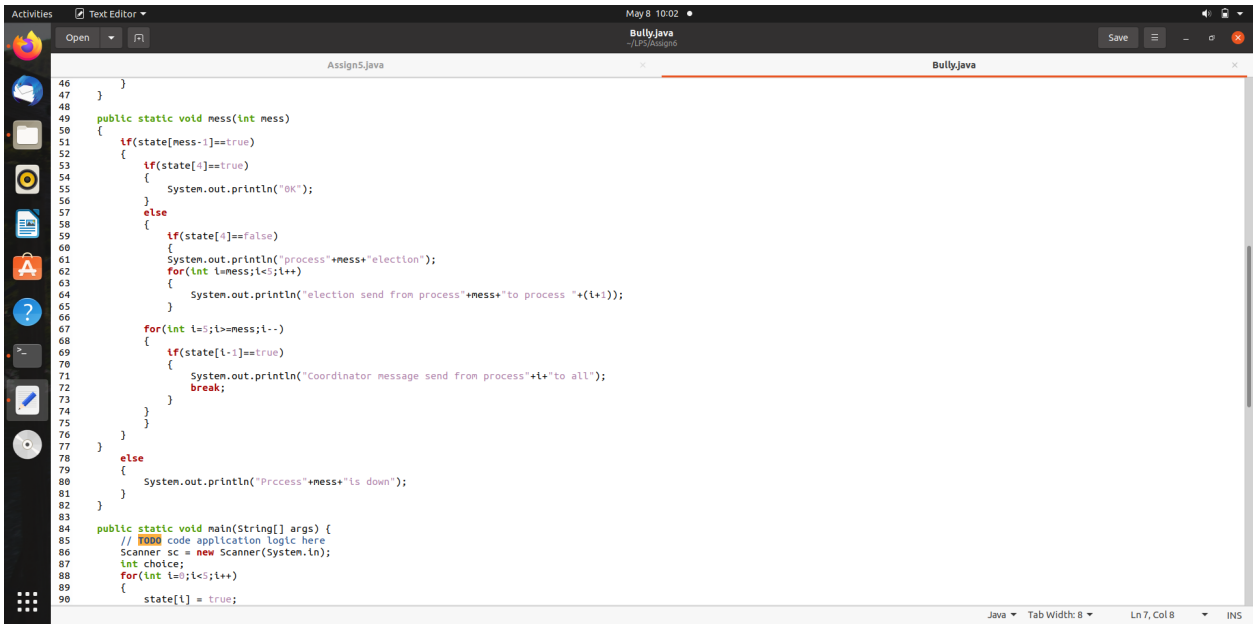
Name- Vaibhav Bichave

Roll no.-43209

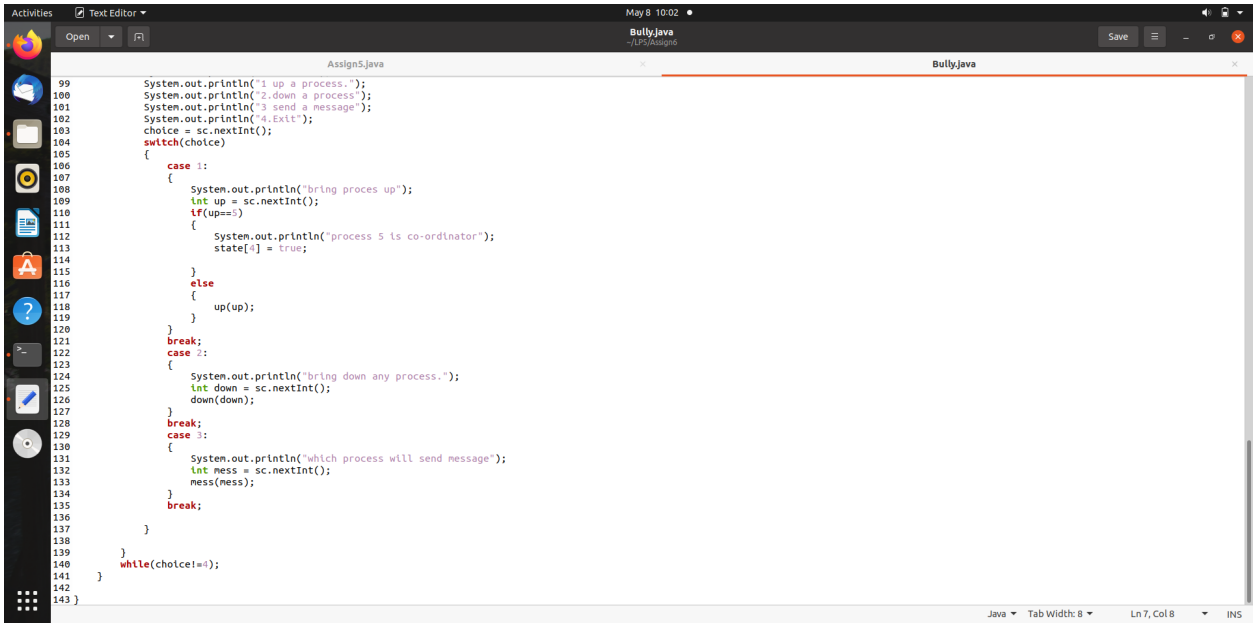
CODE IMPLEMENTATION :

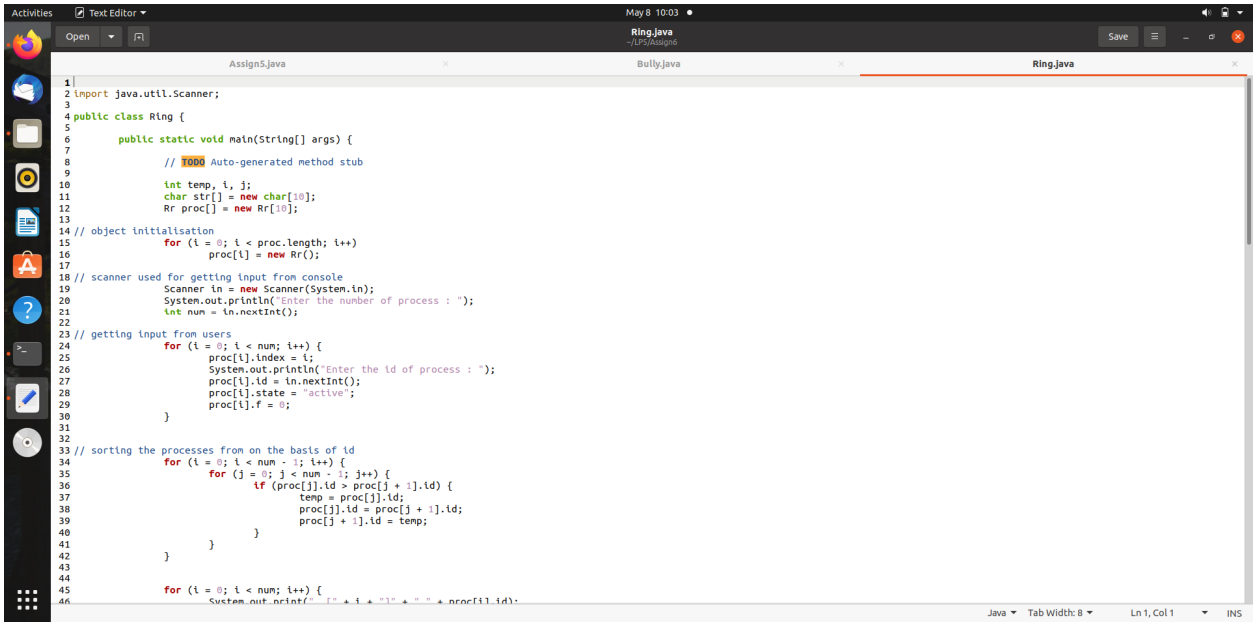


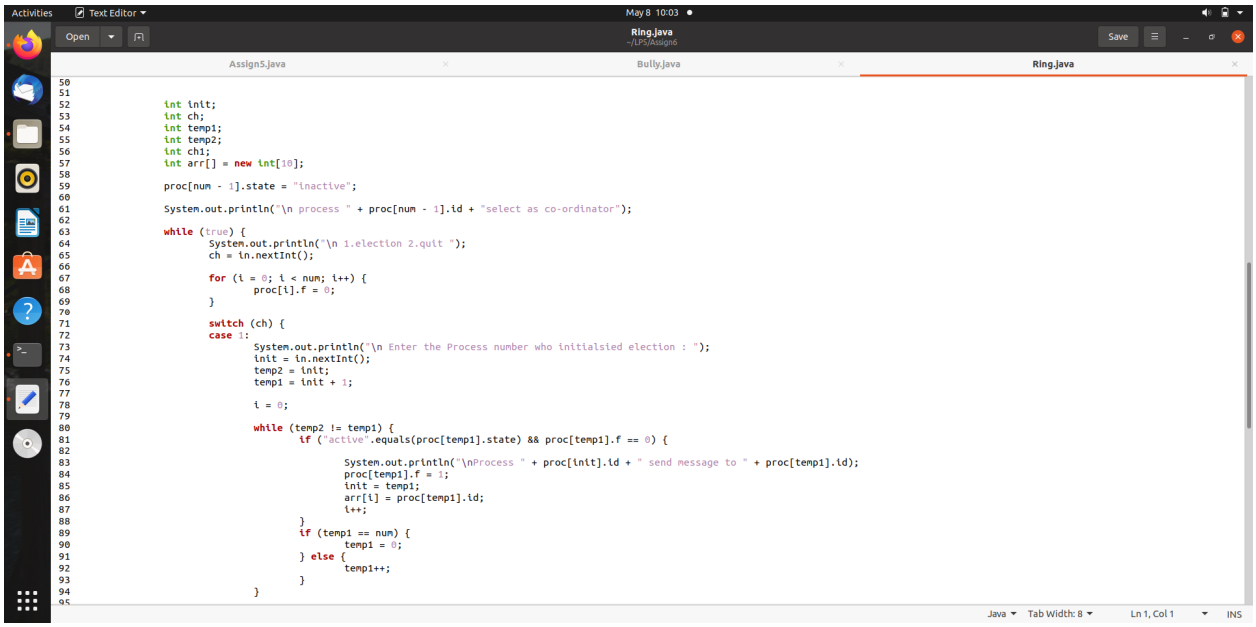
```java
//package bully;

import java.util.Scanner;

public class Bully {

    /**
     * @param args the command line arguments
     */
    static boolean state[]  = new boolean[5];
    int coordinator;
    public static void up(int up)
    {
        if(state[up-1]==true)
        {
            System.out.println("process"+up+"is already up");
        }
        else
        {
            state[up-1] = true;
            System.out.println("process "+up+"held election");
            for(int i=up;i<5;i++)
            {
                System.out.println("election message sent from process"+up+"to process"+(i+1));
            }
            for(int i=up+1;i<=5;i++)
            {
                if(state[i-1]==true)
                {
                   System.out.println("alive message send from process"+i+"to process"+up);
                   break;
                }
            }
        }
    }
    public static void down(int down)
    {
        if(state[down-1]==false)
        {
            System.out.println("process "+down+"is already down.");
        }
        else
        {
            state[down-1] = false;

```

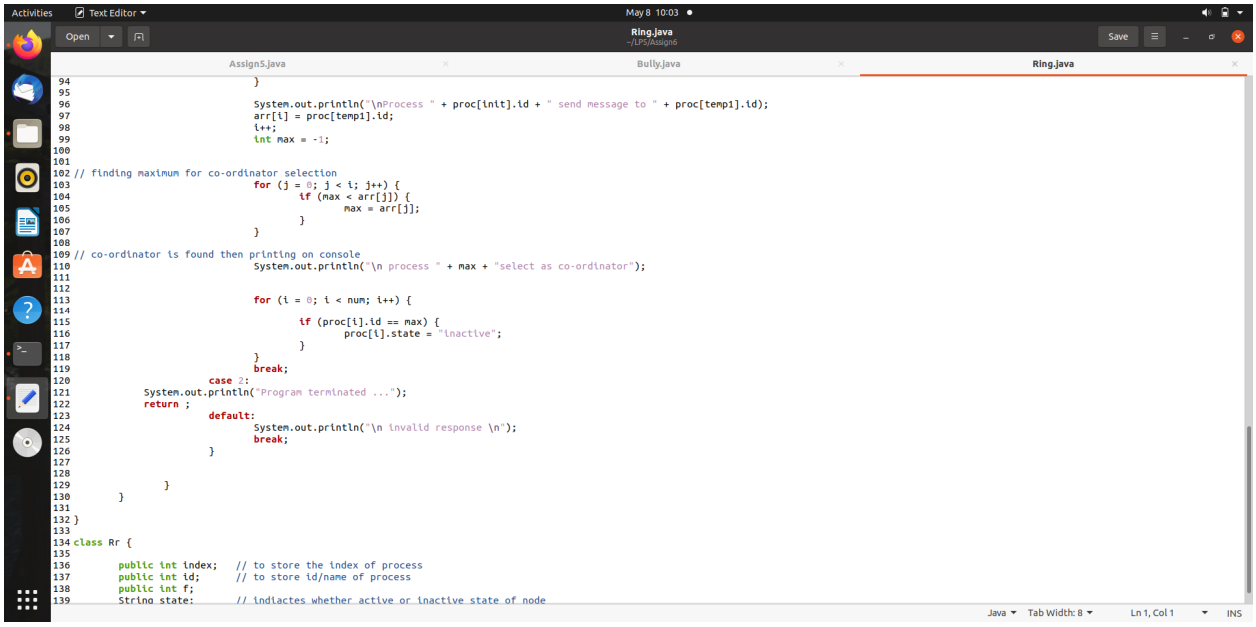


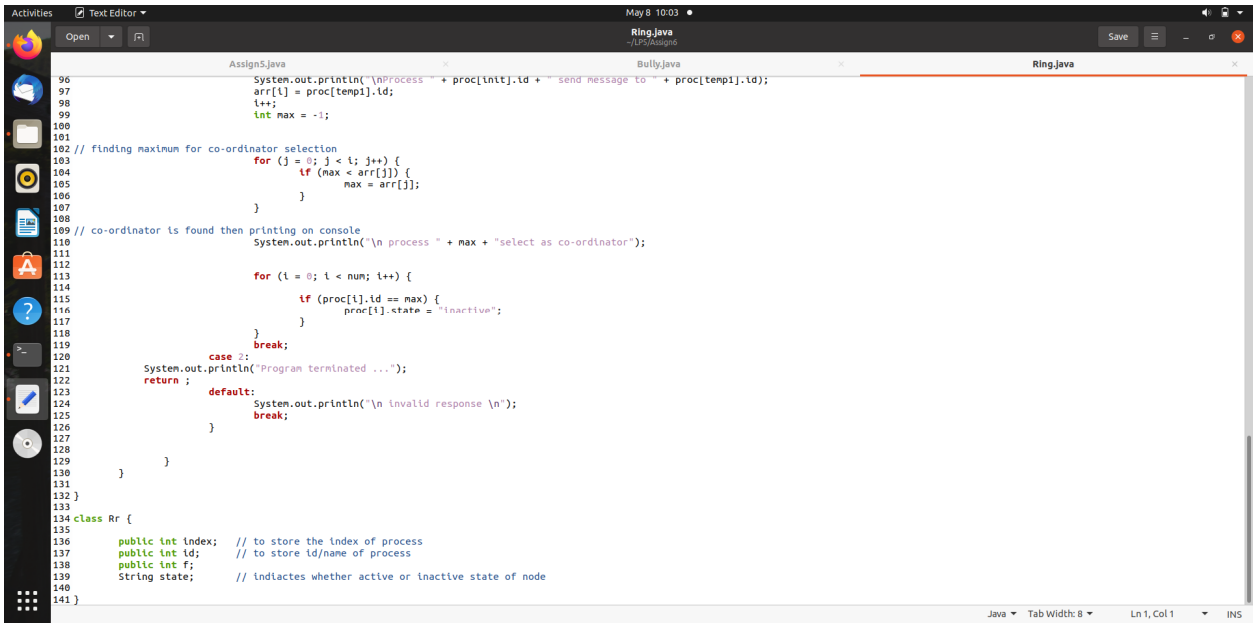
```java
        }
    }

    public static void mess(int mess)
    {
        if(state[mess-1]==true)
        {
            if(state[4]==true)
            {
                System.out.println("OK");
            }
            else
            {
                if(state[4]==false)
                {
                System.out.println("process"+mess+"election");
                for(int i=mess;i<5;i++)
                {
                    System.out.println("election send from process"+mess+"to process "+(i+1));
                }

            for(int i=5;i>=mess;i--)
            {
                if(state[i-1]==true)
                {
                    System.out.println("Coordinator message send from process"+i+"to all");
                    break;
                }
            }
            }
        }
    }
        else
        {
            System.out.println("Prccess"+mess+"is down");
        }
    }

    public static void main(String[] args) {
        // TODO code application logic here
        Scanner sc = new Scanner(System.in);
        int choice;
        for(int i=0;i<5;i++)
        {
            state[i] = true;
```

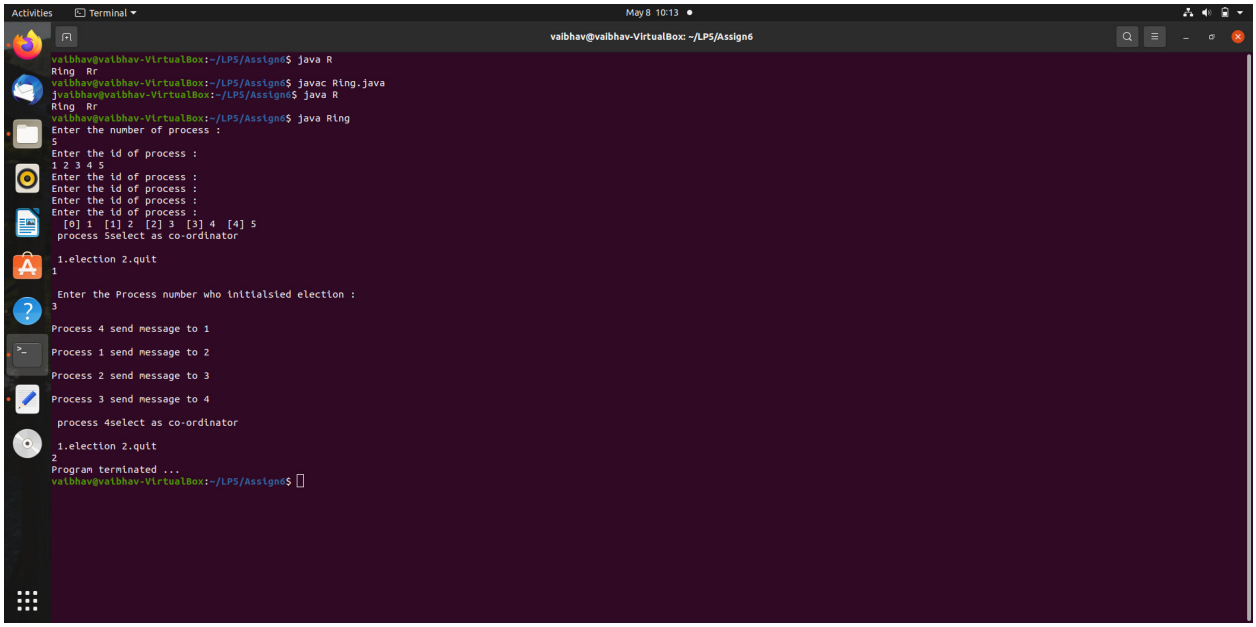

Activities
Text Editor
May 8 10:02
Open
Bully.java
~/LPS/Assign6
Save
Assign5.java
Bully.java
System.out.println("1 up a process.");
System.out.println("2.down a process");
System.out.println("3 send a message");
System.out.println("4.Exit");
choice = sc.nextInt();
switch(choice)
{
case 1:
{
System.out.println("bring proces up");
int up = sc.nextInt();
if(up==5)
{
System.out.println("process 5 is co-ordinator");
state[4] = true;
}
else
{
up(up);
}
}
break;
case 2:
{
System.out.println("bring down any process.");
int down = sc.nextInt();
down(down);
}
break;
case 3:
{
System.out.println("which process will send message");
int mess = sc.nextInt();
mess(mess);
}
break;
}
}
while(choice!=4);
}
}
Java
Tab Width: 8
Ln 7, Col 8
INS

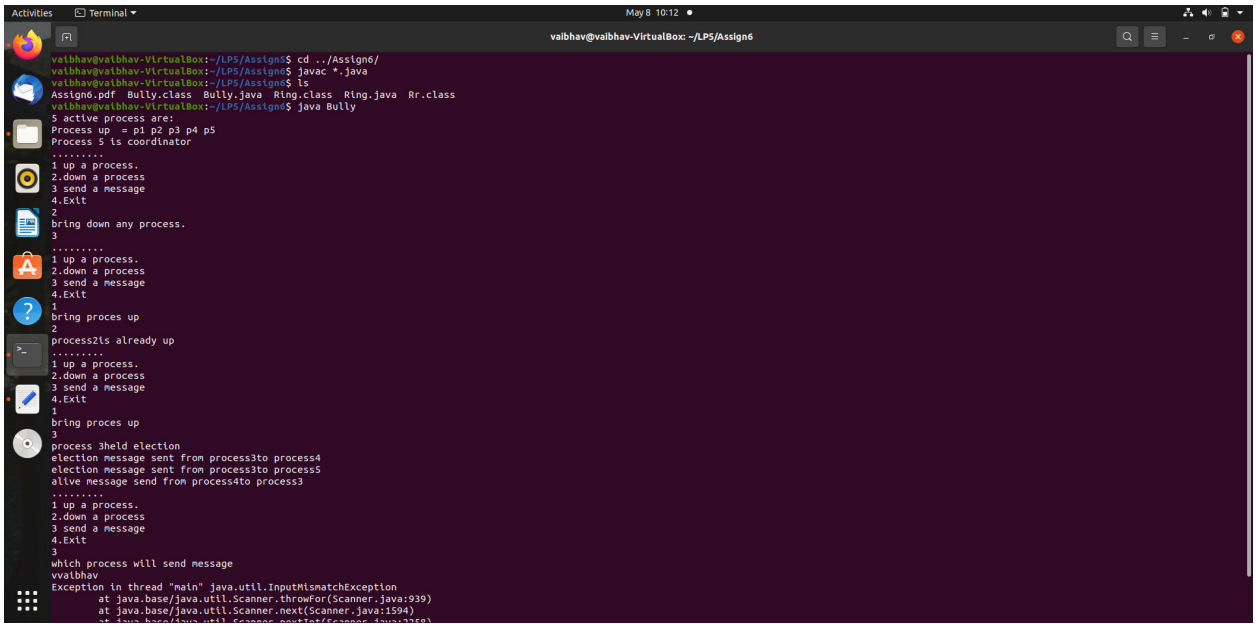

Activities
Text Editor
May 8 10:03
Open
Ring.java
~/LPS/Assign6
Save
Assign5.java
Bully.java
Ring.java
import java.util.Scanner;
public class Ring {
public static void main(String[] args) {
// TODO Auto-generated method stub
int temp, i, j;
char str[] = new char[10];
Rr proc[] = new Rr[10];
// object initialisation
for (i = 0; i < proc.length; i++)
proc[i] = new Rr();
// scanner used for getting input from console
Scanner in = new Scanner(System.in);
System.out.println("Enter the number of process : ");
int num = in.nextInt();
// getting input from users
for (i = 0; i < num; i++) {
proc[i].index = i;
System.out.println("Enter the id of process : ");
proc[i].id = in.nextInt();
proc[i].state = "active";
proc[i].f = 0;
}
// sorting the processes from on the basis of id
for (i = 0; i < num - 1; i++) {
for (j = 0; j < num - 1; j++) {
if (proc[j].id > proc[j + 1].id) {
temp = proc[j].id;
proc[j].id = proc[j + 1].id;
proc[j + 1].id = temp;
}
}
}
for (i = 0; i < num; i++) {
Java
Tab Width: 8
Ln 1, Col 1
INS


Activities
Text Editor
May 8 10:03
Open
Ring.java
~/LPS/Assign6
Save
Assign5.java
Bully.java
Ring.java
int init;
int ch;
int temp1;
int temp2;
int ch1;
int arr[] = new int[10];
proc[num - 1].state = "inactive";
System.out.println("\n process " + proc[num - 1].id + "select as co-ordinator");
while (true) {
System.out.println("\n 1.election 2.quit ");
ch = in.nextInt();
for (i = 0; i < num; i++) {
proc[i].f = 0;
}
switch (ch) {
case 1:
System.out.println("\n Enter the Process number who initialsied election : ");
init = in.nextInt();
temp2 = init;
temp1 = init + 1;
i = 0;
while (temp2 != temp1) {
if ("active".equals(proc[temp1].state) && proc[temp1].f == 0) {
System.out.println("\nProcess " + proc[init].id + " send message to " + proc[temp1].id);
proc[temp1].f = 1;
init = temp1;
arr[i] = proc[temp1].id;
i++;
}
if (temp1 == num) {
temp1 = 0;
} else {
temp1++;
}
}
Java
Tab Width: 8
Ln 1, Col 1
INS


Activities
Text Editor
May 8 10:03
Open
Ring.java
~/LPS/Assign6
Save
Assign5.java
Bully.java
Ring.java
}
System.out.println("\nProcess " + proc[init].id + " send message to " + proc[temp1].id);
arr[i] = proc[temp1].id;
i++;
int max = -1;
// finding maximum for co-ordinator selection
for (j = 0; j < i; j++) {
if (max < arr[j]) {
max = arr[j];
}
}
// co-ordinator is found then printing on console
System.out.println("\n process " + max + "select as co-ordinator");
for (i = 0; i < num; i++) {
if (proc[i].id == max) {
proc[i].state = "inactive";
}
}
break;
case 2:
System.out.println("Program terminated ...");
return ;
default:
System.out.println("\n invalid response \n");
break;
}
}
}
}
class Rr {
public int index; // to store the index of process
public int id; // to store id/name of process
public int f;
String state; // indiactes whether active or inactive state of node
Java
Tab Width: 8
Ln 1, Col 1
INS


Activities
Text Editor
May 8 10:03
Open
Ring.java
~/LP5/Assign6
Save
Assign5.java
Bully.java
Ring.java
System.out.println("\nProcess " + proc[init].id + " send message to " + proc[temp1].id);
arr[i] = proc[temp1].id;
i++;
int max = -1;
// finding maximum for co-ordinator selection
for (j = 0; j < i; j++) {
if (max < arr[j]) {
max = arr[j];
}
}
// co-ordinator is found then printing on console
System.out.println("\n process " + max + "select as co-ordinator");
for (i = 0; i < num; i++) {
if (proc[i].id == max) {
proc[i].state = "inactive";
}
}
break;
case 2:
System.out.println("Program terminated ...");
return ;
default:
System.out.println("\n invalid response \n");
break;
}
}
}
class Rr {
public int index; // to store the index of process
public int id; // to store id/name of process
public int f;
String state; // indiactes whether active or inactive state of node
}
Java
Tab Width: 8
Ln 1, Col 1
INS


Activities
Terminal
May 8 10:13
vaibhav@vaibhav-VirtualBox: ~/LP5/Assign6
vaibhav@vaibhav-VirtualBox:~/LP5/Assign6$ java R
Ring Rr
vaibhav@vaibhav-VirtualBox:~/LP5/Assign6$ javac Ring.java
jvaibhav@vaibhav-VirtualBox:~/LP5/Assign6$ java R
Ring Rr
vaibhav@vaibhav-VirtualBox:~/LP5/Assign6$ java Ring
Enter the number of process :
5
Enter the id of process :
1 2 3 4 5
Enter the id of process :
Enter the id of process :
Enter the id of process :
Enter the id of process :
[0] 1 [1] 2 [2] 3 [3] 4 [4] 5
process 5select as co-ordinator
1.election 2.quit
1
Enter the Process number who initialsied election :
3
Process 4 send message to 1
Process 1 send message to 2
Process 2 send message to 3
Process 3 send message to 4
process 4select as co-ordinator
1.election 2.quit
2
Program terminated ...
vaibhav@vaibhav-VirtualBox:~/LP5/Assign6$


Activities
Terminal
May 8 10:12
vaibhav@vaibhav-VirtualBox: ~/LP5/Assign6
vaibhav@vaibhav-VirtualBox:~/LP5/Assign5$ cd ../Assign6/
vaibhav@vaibhav-VirtualBox:~/LP5/Assign6$ javac *.java
vaibhav@vaibhav-VirtualBox:~/LP5/Assign6$ ls
Assign6.pdf Bully.class Bully.java Ring.class Ring.java Rr.class
vaibhav@vaibhav-VirtualBox:~/LP5/Assign6$ java Bully
5 active process are:
Process up = p1 p2 p3 p4 p5
Process 5 is coordinator
....
1 up a process.
2.down a process
3 send a message
4.Exit
2
bring down any process.
3
....
1 up a process.
2.down a process
3 send a message
4.Exit
1
bring proces up
2
process2ls already up
....
1 up a process.
2.down a process
3 send a message
4.Exit
1
bring proces up
3
process 3held election
election message sent from process3to process4
election message sent from process3to process5
alive message send from process4to process3
....
1 up a process.
2.down a process
3 send a message
4.Exit
3
which process will send message
vvaibhav
Exception in thread "main" java.util.InputMismatchException
at java.base/java.util.Scanner.throwFor(Scanner.java:939)
at java.base/java.util.Scanner.next(Scanner.java:1594)