2024/02/13

Date
Page

Lab-program-10

Open Ended Exercise

* Demonstrate Inter process communication and deadlock.

i) IPC program

```
Class Q{
      Int n;
      boolean valueset=False;
      Synchronized int get(){
      While(!valueset)
      try{
         System.out.println("In consumer
                        waiting\n");
         wait();
      }
       catch(InterruptedException e){
         System.out.println("InterruptedException
                        caught");
       }
     System.out.println("Got:"+n);
      Valueset=False;
      System.out.Println("\nIntimate producer\n");
       notify();
```

```
return n;
}
synchronized void put(int n) {
    while (valueset)
    try {
  System.out.println("\n producer waiting
  wait();
} catch (InterruptedException e) {
System.out.println("InterruptedException
                   caught");
}
 this.n=n;
Value set = true;
System.out.println("put:"+n);
System.out.println("\n Intimate consumer
notify();
}
}
class producer implements Runnable{
Q q;
producer(Q q) {
this.q=q;
new Thread(this,"producer"). start();
}
public void run() {
```

```
int i=0;
While (i<15) {
 q.put( i++);
 }
 }
}

Class consumer implements Runnable{
Q q;
 Consumer (Q q) {
 this.q = q;
 new thread ( this, "consumer").start( );
 }
 public void run( ) {
 int i=0;
 While (i<15) {
 int r=q.get( );
 System.out.println("consumed :"+r);
  i++;
 }
 }
 Class Ipc {
 public static void main(string args[]){
  Q q =new Q( );
  new producer (q);
  new consumer (q);
```

```
        System.out.println("press Control-c to
                                        stop");
    }
}
```

Output:

```
Intimate consumer
Produces waiting

press control-c to stop.
  Got:0

Intimate producer
consumed:0
  Put:1

Intimate consumer
Produces waiting
  Got:1
Intimate producer
consumed:1
  Put:2
```

Intimate producer
~~Consumed 2~~
producer waiting
Got 2
Intimate producer
consumed:2
put:3

Intimate consumer
producer waiting

Got:-3
Intimate ~~consumer~~ producer.
~~Got~~ consumed:3
put:4

Intimate consumer
producer waiting
Got:4
Intimate consumer
consumed:4
Put:5

Date ____
Page ____

Q. (ii) Deadlock program

```
Class A {
    Synchronized void Foo(B b) {
    String name =
    Thread.currentThread().getName();
   System.out.println(name + "entered A.Foo");
   try {
   Thread.sleep(1000);
  } catch(Exception e) {
 System.out.println(name + "trying to call
                        B.last()");
   b.last();
  }
   void last() {
  System.out.println("inside A.last");
 }
}

Class B {
   Synchronized void bar(A a) {
   String name = Thread.currentThread().
                 get.Name().
 System.out.println(name + "entered B.bar
  try {
  Thread. sleep(1000);
```

```
} catch(Exception e){
System.out.println("B Interrupted");
}
System.out.println(name+"trying to call A.
                          last()");
a.last();
}
void last(){
 System.out.println("Inside A.last");
}
}
class Deadlock implements Runnable
{
 A a= new A();
 B b= new B();
 Deadlock(){
 Thread.currentThread().setName("main
                        Thread");
Thread t=new Thread(this,"Racing
                       Thread");
 t.start();
 a.Foo(b); // get lock on a in this
                 thread.
public void run(){
 b.bar(a);
```

classmate
Date
Page

```
        System.out.println("Back in other
                              thread");
    }
    public static void main(String args[]){
    new Deadlock();
    }
}
```

Output :

```
Main Thread entered A.Foo
Racing Thread entered B.bar
Main thread trying to call B.last()
Inside A.last.
Back in main thread.
Racing Thread trying to call. A last()
Inside A.last.
Back in other thread.
```

[illegible] 13/2/2024