

13/02

LAB - 10

Demonstrate Inter process Communication and Deadlock

(i) Interprocess Communication

class Q {

int n;

boolean valueSet = false;

synchronized int get()

{ while (!valueSet)

{ try

{ System.out.println("\n Consumer waiting\n");

wait();

} catch (InterruptedException e) {

System.out.println("got: " + n);

valueSet = true;

System.out.println("\n Intimate Producer\n");

notify();

return n;

}

synchronized void put (int n)

{ while (valueSet)

{ try

{ System.out.println("\n Producer waiting\n");

wait();

} catch (InterruptedException e)

{ System.out.println("InterruptedException caught");

}

this.n = n;

valueSet = true;

System.out.println("Put: " + n);

System.out.println("\n Intimate Consumer\n");

```
3  
3  
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++);
```

```
    }
```

```
}
```

```
3  
3  
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 x = q.get();
```

```
        System.out.println("Consumed: " + x);
```

```
        i++;
```

```
    }
```

```
}
```

class PCFixed

<

public static void main (String args[])

<

q = new q();

new Producer(q);

new Consumer(q);

System.out.println (" Press Control-C to stop");

}

}

OUTPUT

Put : 0

Intimate Consumer

Producer Waiting

Press Control-C to stop

Got : 0

Intimate Producer

Consumed : 0

Put : 1

Intimate Consumer

Producer Waiting

Got : 1

Intimate Producer

Consumed : 1

Put : 2

Intimate Consumer

Producer waiting

Got : 2

Intimate Producer

Consumed : 2

Put : 3

SANTANA SURESH

30M22CS239

Intimate consumer
Producer waiting

got : 1

Intimate consumer Producer

~~Producer waiting~~ consumed : 1

got :

Put : 4

Intimate consumer

Producer waiting

got : 4

Intimate Producer

consumed : 4

Put : 5

Intimate consumer

Producer waiting

got : 5

Intimate Producer

consumed : 5

Put : 6

Intimate consumer

Producer waiting

got : 6

Intimate Producer

consumed : 6

Put : 7

Intimate consumer

~~Producer waiting~~

got : 7

Intimate Producer

consumed : 7

Put : 8

Intimate consumer

SANTANA SURESH

18M92CS239

Producer waiting

got : 8

Intimate Producer

consumed : 8

Put : 9

Intimate Consumer

Producer waiting

got : 9

Intimate Producer

consumed : 9

Put : 10

Intimate Consumer

Producer waiting

got : 10

Intimate Producer

consumed : 10

Put : 11

Intimate Consumer

Producer waiting

got : 11

Intimate Producer

consumed : 11

Put : 12

Intimate Consumer

Producer waiting

got : 11

Intimate Producer

consumed : 11

Put : 13

SANTANA SURESH

10M22 CS239

Producer writing

GA: 12

Intimate Producer

answered = 12

Put = 12

Intimate answer

Producer writing

SANTANA SRESH

2AM 22CS279

was
13-2-24

(b) Deadlock

class A

↳ synchronized void foo (B b)

↳ String name = Thread.currentThread().getName();
System.out.println (name + " entered A.foo");

try

↳ Thread.sleep (1000);

3 catch (Exception e)

↳ System.out.println ("A interrupted");

3

System.out.println (name + " trying to call B.last()");

b.last();

3

void last()

↳ System.out.println ("Inside A's last");

3

3

class B

↳ synchronized void bar (A a)

↳ String name = Thread.currentThread().getName();

System.out.println (name + " entered B.bar");

try

↳ Thread.sleep (1000);

3 catch (Exception e)

↳ System.out.println ("B interrupted");

3

System.out.println (name + " trying to call A.last()");

2. last ()

3

void last ()

1 System.out.println (" Inside A: last ");

3

3

class Deadlock implements Runnable

1 A a = new A ();

B b = new B ();

Deadlock ()

1

Thread.currentThread (). setName (" Main Thread ");

Thread t = new Thread (this, " Racing Thread ");

t.start ();

a.run (b);

System.out.println (" Back in main thread ");

3

public void run ()

1

b.run (a);

System.out.println (" Back in another thread ");

3

public static void main (String args [])

1

new Deadlock ();

3

3

o/p

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

SANTANA SURESH

18M02CS231

13-2-24