Practical 3

***AIM:*** Synchronization: a. Write a program to give a solution to the Bounded buffer problem. b. Write a program to give a solution to the readers–writers problem.

***CODE:***

import java.util.concurrent.Semaphore;

class Q{

// an item

int item;

// semCon initialized with 0 permits

// to ensure put() executes first

static Semaphore semCon = new Semaphore(0);

static Semaphore semprod = new Semaphore(1);

// to get an item from buffer

void get(){

try{

// Before consumer can consume an item

// it must aquire a permit from semCon

semCon.acquire();

}

catch(InterruptedException e){

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

}

// consumer consuming an item

System.out.println("\n Consumer consumed item :"+item);

// After consumer consumes the item

// It releases semProd to notify producer

semprod.release();

}

// to put an item in buffer

void put(int item){

try{

// Before producer can produce an item

// it must acquire a permit from semprod

semprod.acquire();

}

catch(InterruptedException e){

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

}

// producer producing an item

this.item = item;

System.out.println("\n Producer produced item :"+item);

// After producer produces the item

// it releases semcon to notify consumer

semCon.release();

}

}

// Producer class

class producer implements Runnable{

Q q;

producer(Q q){

this.q = q;

new Thread(this,"producer").start();

}

@Override

public void run(){

for(int i=0; i<5; i++)

// Producer put items

q.put(i);

}

}

// consumer class

class consumer implements Runnable{

Q q;

consumer(Q q){

this.q = q;

new Thread(this,"consumer").start();

}

@Override

public void run(){

for(int i=0; i<5; i++)

// Consumer put items

q.get();

}

}

// Driver class

public class PT {

public static void main(String[] args){

// creating buffer queue

Q q = new Q();

// Starting consumer thread

new consumer(q);

// Starting producer thread

new producer(q);

}

}

**OUTPUT:**

Producer produced item :0

Consumer consumed item :0

Producer produced item :1

Consumer consumed item :1

Producer produced item :2

Consumer consumed item :2

Producer produced item :3

Consumer consumed item :3

Producer produced item :4

Consumer consumed item :4