import

java.util.concurrent.Semaphore;

public class ReaderWriter {

static int rc=0;

static Semaphore s = new Semaphore(1);

static Semaphore wrt= new Semaphore(1);

public static void main(String[] args) {

Reader r = new Reader();

Writer w= new Writer();

Thread t1= new Thread(r);

Thread t2= new Thread(r);

Thread t3= new Thread(r);

Thread t4= new Thread(w);

Thread t5= new Thread(w);

Thread t6= new Thread(w);

t1.setName("Reader1");

t2.setName("Reader2");

t4.setName("Writer1");

t3.setName("Reader3");

t5.setName("Writer2");

t6.setName("Writer3");

t1.start();

t2.start();

t3.start();

t4.start();

t5.start();

t6.start();

}

static public class Reader implements Runnable //Reader Program

{

public void run()

{

try {

s.acquire();

rc++;

if(rc==1)

{

wrt.acquire();

}

s.release();

System.out.println(Thread.currentThread().getName()+"is Reader"); // Prints the current reader

Thread.sleep(1000);

System.out.println("read Finish");

s.acquire();

rc--;

if(rc==0)

{

wrt.release();

}

s.release();

}catch (InterruptedException e)

{

System.out.println(" Error!! try again ");

}

}

}

static public class Writer implements Runnable //Writer Program

{

public void run()

{

try {

wrt.acquire();

{

System.out.println(Thread.currentThread().getName()+"writer is writing");// print the currnt writer

Thread.sleep(1000);

}

wrt.release();

}catch (InterruptedException e)

{

System.out.println(" Error!! try again ");

}

}

}

}

when reader is reading:



When reader has finished reading , i.e. it starts writing :

