

Name: Sayyed Sohail Rashid	Course Name: DC-LAB
Class: BE-CO	Batch: 01
Roll no: 18CO48	Experiment No: 06

Aim : To Implement the Deadlock Detection Algorithm.

Code:

TestThread.java

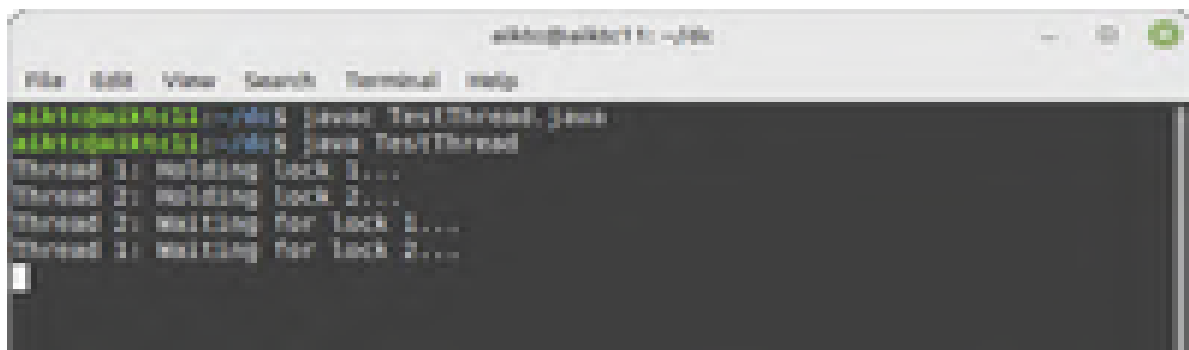
```
import java.io.*;
public class TestThread {
    public static Object Lock1 = new Object();
    public static Object Lock2 = new Object();

    public static void main(String args[]){
        ThreadDemo1 T1 = new ThreadDemo1();
        ThreadDemo2 T2 = new ThreadDemo2();
        T1.start();
        T2.start();
    }

    private static class ThreadDemo1 extends Thread {
        public void run() {
            synchronized (Lock1) {
                System.out.println("Thread 1: Holding lock 1...");
                try{ Thread.sleep(10); }
                catch(InterruptedException e){}
                System.out.println("Thread 1: Waiting for lock 2...");
                synchronized(Lock2){
                    System.out.println("Thread 1: Holding lock 1 & 2...");
                }
            }
        }
    }

    synchronized(Lock1){
        System.out.println("Thread 2: Holding lock 1 & 2...");
    }
}
```

Output:



```
aditya@aditya: ~ % java TestThread, Java
aditya@aditya: ~ % java TestThread
Thread 1: holding lock 1...
Thread 2: waiting lock 2...
Thread 2: waiting for lock 1...
Thread 1: waiting for lock 2...
```

The screenshot shows a terminal window with a dark background. The title bar reads 'aditya@aditya: ~ %'. The terminal displays the execution of a Java program. The first line shows the command 'java TestThread, Java' being executed. The second line shows the command 'java TestThread' being executed. The output consists of four lines: 'Thread 1: holding lock 1...', 'Thread 2: waiting lock 2...', 'Thread 2: waiting for lock 1...', and 'Thread 1: waiting for lock 2...'. The program has reached a deadlock state where both threads are waiting for a lock held by the other.

Conclusion:

Deadlock has been successfully detected between the two threads.