I’m including the code here for ease-of-reference.

Note that **this is basically the example given in Section 6.5.3 of the book (on page 254)**

If you want to run it yourself: This should run (either unmodified or with a couple of small changes). You’ll know that the processes have deadlocked when they stop producing output (at which point you can have the debugger pause the process in order to examine which thread is where)

|  |  |
| --- | --- |
| Line | Code |
|  | import java.util.concurrent.Semaphore; |
|  | public class DeadLockExample { |
|  |  |
|  |  public static void Main() { |
|  |  Semaphore S = new Semaphore(1); // use mutex-style Semaphores |
|  |  Semaphore Q = new Semaphore(1); |
|  |  MyThread t1 = new MyThread(true, S, Q); |
|  |  MyThread t2 = new MyThread(false, S, Q); |
|  |  |
|  |  (new Thread(t1)).start(); |
|  |  (new Thread(t2)).start(); |
|  |  |
|  |  // Now wait for deadlock :) |
|  |  } |
|  | } |
|  |  |
|  | class MyThread implements Runnable { |
|  |  Semaphore S; |
|  |  Semaphore Q; |
|  |  boolean s\_first; |
|  |  |
|  |  public MyThread(boolean which, Semaphore arg\_S, Semaphore arg\_Q) { |
|  |  s\_first = which; |
|  |  S = arg\_S; // naming args with "arg\_" is goofy, but |
|  |  Q = arg\_Q; // this.Q = Q makes me nervous :) |
|  |  } |
|  |  |
|  |  @Override |
|  |  public void run() { |
|  |  try { |
|  |  while (true) { |
|  |  if( s\_first ) { |
|  |  System.out.println(Thread.currentThread().getName()+": About to acquire"); |
|  |  S.acquire(); |
|  |  Q.acquire(); |
|  |  S.release(); |
|  |  Q.release(); |
|  |  } else { |
|  |  System.out.println(Thread.currentThread().getName()+": About to acquire"); |
|  |  Q.acquire(); |
|  |  S.acquire(); |
|  |  Q.release(); |
|  |  S.release(); |
|  |  } |
|  |  } |
|  |  } |
|  |  catch( InterruptedException ie){ |
|  |  System.out.println("Got an interrupted exception!" + ie.getMessage()); |
|  |  ie.printStackTrace(); |
|  |  } |
|  |  } |
|  | } |