Socket通信中对象流阻塞问题研究

Socket编程中一个很奇怪的现象：用DataInputStream对象没错，但改为ObjectInputStream就出错了，而且try...catch还抓不到异常。  
  
例如在Socket客户端中，使用DataInputStream进行下述编程，很正常，不会出现问题：

|  |
| --- |
| DataInputStream dis = new DataInputStream(socket.getInputStream); DataOutputStream dos = new DataOutputStream(socket.getOutputStream); |

但若改用ObjectInputStream对象进行编程，如下：

|  |
| --- |
| ObjectInputStream ois = new ObjectInputStream(socket.getInputStream);  ObjectOutputStream oos = new ObjectOutputStream(socket.getOutputStream); |

这时程序就死锁住了，不报错或异常出来，用try...catch也抓不到异常信息。

到网络传递对象时，遇到了一个很奇怪的问题。代码执行到

new ObjectInputStream(socket.getInputStream());

就执行不下去了。

问题原因： 原因是建立ObjectInputStream对象是需要先接收一定的header数据，接收到这些数据之前会处于阻塞状态。而这个header数据是由对方的ObjectOutputStream创建的时候发出来的。所以如果只有单方面的ObjectInputStream一定会缺少header数据导致卡死。

解决方法：

Server、Client两端同时建立对象流， 约定好创建顺序， ：

Server:

socket = ss.accept()

ObjectInputStream ois = new ObjectInputStream(socket.getInputStream()); (第一步)

ObjectOutputStream oos = new ObjectOutputStream(socket.getOutputStream());

Client:

Socket socket = new Socket("10.0.2.2", 10000);

socket.setSoTimeout(10000);

ObjectOutputStream oos = new ObjectOutputStream(socket.getOutputStream()); (第一步)

ObjectInputStream ois = new ObjectInputStream(socket.getInputStream());