BIO的同步机制最好双方的读写一致，也就是怎么写怎么读。

（连接重置异常原因：

客户端数据发送完成而服务端还在等待数据，而当客户端断连接，服务端套接字检查到对方已经死了，这边就殉情，于是该通道就会返回给输入输出流异常，输入输出流再讲异常抛出

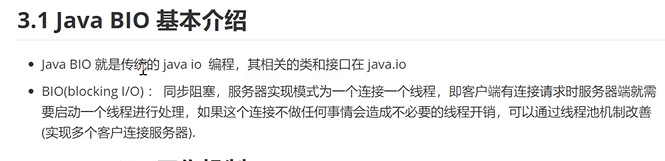


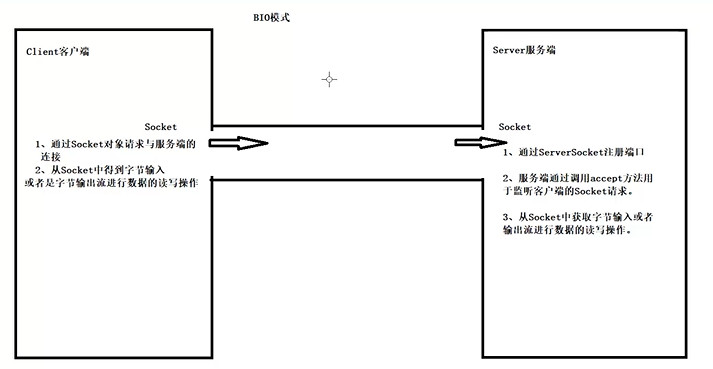
终究其原因是因为服务端不知道数据已经发送完毕，而错误的调用读写操作。所以使用一个方法通知服务端数据发送完毕。

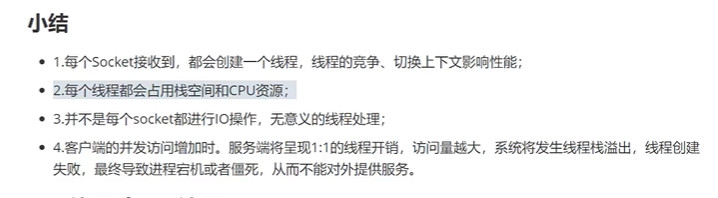
另有验证客户端关闭流就能让客户端知道消息结束，并不需要关闭socket。

BIO算的上是一个长连接。

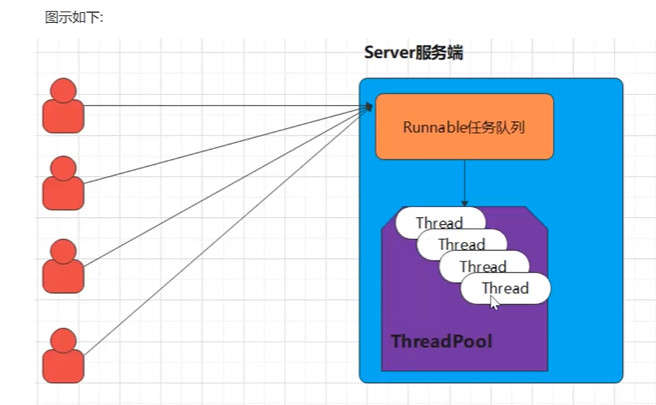
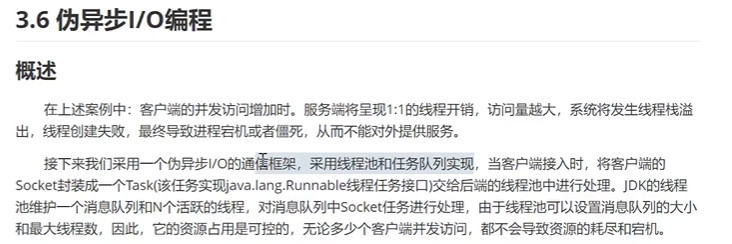
）







**伪异步：**



**消息队列：（发送消息的源头与消息的终点之间暂存消息的容器，比如QQ，如果你向一个不在线的好友发送消息，他没有登录，那么就无法直接收到消息，保存在消息队列中直到某天他上线）**

[**https://baike.baidu.com/item/%E6%B6%88%E6%81%AF%E9%98%9F%E5%88%97/4751675?fr=aladdin**](https://baike.baidu.com/item/%E6%B6%88%E6%81%AF%E9%98%9F%E5%88%97/4751675?fr=aladdin)

**线程池参数详解：**

[Java线程池七个参数详解\_IT小跟班-CSDN博客\_线程池参数详解](https://blog.csdn.net/ye17186/article/details/89467919)

<https://blog.csdn.net/ye17186/article/details/89467919>

**（流程：**

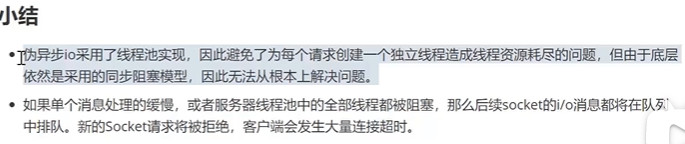
**1.服务端建立一个线程池**

**2.服务端接收一个客户端连接，得到一个套接字**

**3.讲套接字封装为一个任务，注意任务（Task）的实现了Runnable接口，所以可以直接做一个Runnable实现类作为任务类，然后提交给线程池**

**4.提交给线程，线程池进行执行**

**）**



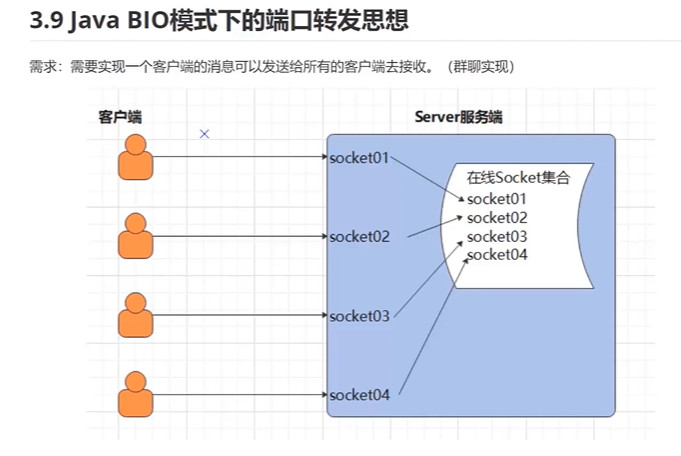
**BIO上传下载任意类型的文件**

**（DataOutputSream分段发送输出流，可做到先发送一段字符串，再发送具体的文件数据）**





**BIO模式下接口转发思想：**



**（比如QQ端到端的转发，并不是直接的P2P通信而是通过服务器进行转接）**

**（端口转发字节：客户端先将字节发送到服务端，再由服务端转发到其他客户端）**

（服务端每次得到一个socket都会存到一个在线socket中去）

（客户端需要线程发送信息，另一个线程接收信息）



