问题： 1、（M90）netty是如何从端口上报数据值accepter在呢。

没有明白问题是什么意思。麻烦再详细说明一下~

2、（F10）请问老师，Java nio基于事件驱动的四个事件，能不能分别解释一下这四个事件的具体情况么，视频里面直接一笔带过了，我不太明白accept和connect的区别，一个是接收到连接的事件和建立连接的事件，一般后者用在什么场景呢？ 另外read可读事件，指的是数据已经到用户态了么，还是说才到核心态，如果是到了核心态的话，为什么我不可以等到用户态再响应呢，这样更可以减少阻塞时间，如果是用户态的话，对应第一节课的io分类，那不就是异步io了，和老师说的nio使用的是io复用有些矛盾，还是说老师说的是io复用指的是数据到用户态的阻塞，以及数据处理的阻塞，这两个过程的复用呢？不知道我想的对不对，非常期待您的解惑。

OP\_ACCEPT事件就绪：当收到一个客户端的连接请求时，该事件就绪。这是ServerSocketChannel上唯一有效的操作。（服务器端）

OP\_CONNECT事件就绪：只有客户端SocketChannel会注册该操作，当客户端调用SocketChannel.connect()时，该事件就绪。（客户端）

OP\_READ事件就绪：该操作对客户端和服务端的SocketChannel都有效，当OS的读缓冲区中有数据可读时，该事件就绪。

OP\_WRITE事件就绪：该操作对客户端和服务端的SocketChannel都有效，当OS的写缓冲区中有空闲的空间时，该事件就绪。

accept和connect的区别：accept是服务器端接收到连接，connect则是客户端发起连接成功。后者用在客户端，当连接成功后注册相关读写事件。

对，read就指数据已经到内核缓冲区了，将其拷贝到应用缓冲区的过程。如果等到用户态再响应就是异步IO了。目前Linux异步IO接口还不成熟，因此使用较少。

IO复用更多的是指一个线程监控多个IO通道，这个监控线程的复用。

3、针对图中这里实现多线程reactor模式，这里多线程是如何配置的呢？

就是在netty程序的ChannelHandler中的channelRead()方法中，创建新的线程（池）来执行处理操作。

4、（K32）如果根据字节长度分割数据流，那么接收到的数据长度不一致怎么办呢？（多个不同类型的物联网设备）

在发送时通过LengthFieldPrepender添加数据长度前缀，然后通过LengthFieldBasedFrameDecoder拆包即可。