问题：

1. channelRead方法是多线程调用吗？（L23）

channelRead方法不会被多线程调用，只会在channel对应的Eventloop所在的线程所调用。

1. 老师一个serversocketchannel可以注册到多个selector上吗？ serversocket只能注册到一个端口，怎么做到eventLoopgroup的？

Serversocketchannel不能注册到多个Selector上去，只会注册在一个Selector上。（注册到多个Selector, accept()在哪个线程上返回了？）

Serversocketchannel绑定一个端口，并且注册到Boss EventloopGroup上去，他也只会使用BossEventloopGroup中的某一个Eventloop，也就是一个线程。

3、ByteBuf 中 array() 在什么场景下能取到值？

bossgroup 如何多线程工作完成accept？（G90）

当ByteBuf.hasArray()返回True时，array()就能够返回底层对应的数组。只有当ByteBuf是HeapByteBuf时才能够返回其底层array，为DirectByteBuf时则是无法取得。

Accept只会在Boss EventLoopGroup中的一个线程上运行，不会在多线程中同时accept。

1. 那个衍生缓冲区和符合缓冲区有什么区别？

衍生缓冲区和复合缓冲区都是一个视图，自身并没有真正的底层存储空间。

衍生缓冲层是一个原缓冲区的视图，与原缓冲区共享底层存储空间，但是维护自己的索引（readerIndex和writerIndex）。

复合缓冲层则是多个缓冲区的一个整体视图。

5、是不是换行符和定长这些方案就能解决半包粘包问题对吧，然后真正项目也是这么用的对吧？

换行符（或基于其他分隔符）和定长是解决半包和粘包的一种方案。在有些项目中有用到。用得最广的应该还是基于长度的解码器，也是就协议中有一个字段指明数据的长度。

1. socket编程，如果是聊天室类型的，页面连接socket，除了websocket之外，还有其他长连接方式么？

网页的话目前主要是Websocket。

7、EventLoopGroup包含多个EventLoop，每个EventLoop包含一个selector,这个selector会管理多个channel，一个channel可以有多个客服端连接（一个Channel只表示一个服务器和客户端连接），

每一个channel都有一个ChannelPipeline,ChannelPipeline包含多个channelHandler实例列表，

每一个channelHandler对应多（一个）channelHandlerContentext，channelHandlerContentext只能对应一个channelHandler

8、 8.1、ChannelPipeline通过高级截取过滤器模式来掌控ChannelHandler处理事件（入站[inbound]和出站[outbound]）的流程；

8.2、ChannelHandler并没有方法处理事件（嗯，只有一些通用的生命周期方法，handlerAdded/handerRemoved等），而需要由子类处理：ChannelInboundHandler拦截和处理入站事件，ChannelOutboundHandler拦截和处理出站事件。

8.3、ChannelHandlerContext，上下文处理，指的是ChannleHandler之间的关系以及ChannelHandler与ChannelPipeline之间的关系，ChannelPipeline中的事件传播主要依赖于ChannelHandlerContext实现，

由于ChannelHandlerContext中有ChannelHandler之间的关系（不是ChannelHandlerContext有ChannelHandler的关系，是各个ChannelHandler的ChannelHandlerContext之间是一条双向链表），所以能得到ChannelHandler的后继节点，从而将事件传播到下一个ChannelHandler。

9、 流程：一个客户端 --》 EventLoop -》 selector -》 channel -》 ChannelPipeline（开始数据处理），

（1）一个channel可以有多个客服端连接，每一个channel都有一个ChannelPipeline，所有可以处理多个客户端连接后的数据处理，（一个Channel只表示一个服务器和客户端连接）

因为Netty提供了Sharable注解，如果一个ChannelHandle状态无关，那么就标注为Sharable，这样就可以实现一个实例处理所有客户端的事件 ；（如果一个ChannelHandle无状态，有Sharable注册，那在所有Channel的ChannelPipeline中就可以使用同一个实例，而不需要为每个channel创建一个单独的实例）

（2）入站[inbound]：处理服务端接受客户端过来的数据；出站[outbound]：处理服务端发送客户端的数据

这样说不准确。你是从服务器程序角度来说的。而对于客户端程序角度来说，则相反：入站处理客户端接受服务端过来的数据；出站则是处理客户端发送服务端的数据。

入站则主要处理从Channel进来的事件，像读；出站主要处理从channel出去的事件，像写，连接。