

谢尊锦:20:37:51

老雷，ChannelInitializer 是每次有连接来创建 NioSocketChannel 后都执行一次

谢尊锦:20:40:59

TailContext 实现 ChannelInboundHandler，而 HeadContext 都实现了 ChannelOutboundHandler, ChannelInboundHandler, 这样设计是基于什么考虑的？

答：在编码过程中 HeadContext 里需要用到 ChannelOutboundHandler 的方法,而 TailContext 用不到.

阿杜:20:42:38

入站

谢尊锦:20:48:31

一般出站是用户发起的

阿杜:20:48:33

in 是读操作，out 是写操作，是吧

谢尊锦:20:48:45

所以不用 tail

牛志刚:20:48:52

头疼。

牛志刚:20:48:55

我个人现在听不懂。

大胜.彼得:20:49:42

又是一个大佬

Ghost:20:58:09

那个位运算的意义是啥子呀啊? 》

答: 做标记.

湮汐:20:58:09

位运算 相当于是 避免写一堆 true false 吗?

soso-李卓:20:59:05

掩码 一个道理

答: 对的.

谢尊锦:21:02:49

嗯, 最底层

Reythor 雷:21:07:35

1149

湮汐:21:08:45

是要等绳子来的时候 才能执行吧

大胜.彼得:21:09:03

客户端数据过来的时候执行?

答: 是的.

谢尊锦:21:10:44

accept 事件后

谢尊锦:21:17:50

为什么每次新连接来都要重新创建 pipeline 和里面的 handler 实例,这样设计对性能影响咋样?

soso-李卓:21:18:48

。 handler

湮汐:21:19:35

handler 可以设置成单例 但是要加上注解@Shareable, 但是有状态的 handler 不能用单例

杨腾飞:21:20:02

地主家的临时工

soso-李卓:21:20:02

只负责加载

soso-李卓:21:26:29

短工被辞退了

阿杜:21:27:14

哈哈, 外包呀

谢尊锦:21:27:37

严格说不是辞退, 而是对每个客户只服务一次

大胜.彼得:21:28:25

把将来真正要处理数据的处理器永久保留在 pipeline, 而负责一次添加的那个外壳处理器被用完就删除

答：理解的很到位。

谢尊锦:21:34:18

好像 ChannelInitializer 只 new 一次

谢尊锦:21:35:51

但可以复用

谢尊锦:21:36:23

没干掉

谢尊锦:21:36:30

只是移除链表

谢尊锦:21:37:00

就还是同一个临时工

soso-李卓:21:37:05

就是辞退了 没整死它

soso-李卓:21:37:09

哈

Mr 陈:21:37:26

临时工也得到处打工啊

soso-李卓:21:43:49

双向链表插入

soso-李卓:21:44:39

大一基础课

黄欣健:21:45:03

大二学

Free3610:21:45:12

大二学的

黄欣健:21:45:13

操作系统大三

湮汐:21:45:13

切一下视频吧

牛志刚:21:45:27

非科班

谢尊锦:21:45:36

大二逃课了以为是大一学的

soso-李卓:21:45:52

有道理 哈

soso-李卓:21:47:23

会被 GC?

soso-李卓:21:47:52

整死了 哈

Mr 陈:21:50:59

不能被回收啊

谢尊锦:21:51:32

它指向别人

谢尊锦:21:51:40

可以回收

杨腾飞:21:51:48

`pre.next.prev = null; prev.next.next = null;`

杨腾飞:22:27:41

simple

湮汐:22:27:44

以前是 Simple

Mr 陈:22:36:40

为什么 2222 开头 而不是 111111

Mr 陈:22:37:06

o o o o o

Mr 陈:22:37:10

哦哦哦

谢尊锦:22:42:53

tail 不需要 outbound 接口，但需要它来找下一个出站节点，  
这样理解吧？

答：把 tail 声明成 outbound 就是为了调 unsafe,找选一个出站  
节点跟这个没关系.

soso-李卓:22:43:31

就是按顺序跑的把

余爽:22:44:53

因为 outbound 是倒序的吧

谢尊锦:22:45:15

对，明白了

Mr 陈:22:45:39

整体还是萌萌哒

大胜.彼得:22:46:43

往前走一步 又往后退一步，死循环吗

soso-李卓:22:47:35

长链接

谢尊锦:22:48:05

如果连接死了 会占用资源

xiaoLee-hangzhou:22:48:55

占用资源

湮汐:22:50:22

老师，IDEA 的“stop”是正常关闭连接，还是“突然死了”？

答：正常停止.

湮汐:22:51:39

杀进程呢？

答：杀进程不是正常停止.

Mr 陈:22:52:09

服务器 经常直接杀死进行啊

soso-李卓:22:52:29

ping pong

Free3610:22:53:56

这块录播好像讲过一点

谢尊锦:23:00:22

异常的传播是从头到尾全部节点触发

答：是的,但是得要 `fireExceptionCaught` 方法,不然是不会向下传的.

大胜.彼得:23:01:39

录播里好像有讲这个

xxzx\_4684222:23:01:52

有说过

xiaoLee-hangzhou:23:02:19

事件触发就是用来断开连接的

soso-李卓:23:02:53

ping pong

xiaoLee-hangzhou:23:03:09

ping pong



soso-李卓:23:03:10

控制包

