

我是一段不羁的公告！
记得给芬芳这 3 个项目加油，添加一个 STAR 噢。
<https://github.com/YunaiV/SpringBoot-Labs>
<https://github.com/YunaiV/onemail>
<https://github.com/YunaiV/ruoyi-vue-pro>

• NETTY

精尽 Netty 源码解析 —— Channel（八）之 disconnect 操作

1. 概述

本文分享 Netty NIO Channel **客户端**断开连接(**disconnect**)操作的过程。

在看 Netty NIO Channel 对 `#disconnect(ChannelPromise promise)` 方法的实现代码之前，我们先来看看 Java **原生** NIO SocketChannel 的 `#disconnect()` 方法。

- 结果，结果，结果，翻了半天，只看到 NIO SocketChannel 的父类 `AbstractInterruptibleChannel` 中，有 `#close()` 方法，而找不到 `#disconnect()` 方法。这个是啥情况？
- 我们又去翻了 Java **原生** `UDP DatagramSocket` 类，结果找到了 `#connect()` 方法。这个又是啥情况？

不卖关子了，直接说结论啦：

- Java **原生** NIO SocketChannel **不存在**，当调用 `Netty NioSocketChannel#disconnect(ChannelPromise promise)` 时，会自动转换成 **close** 操作，即《精尽 Netty 源码解析 —— Channel（七）之 close 操作》。
- 实际上，`Channel#disconnect(ChannelPromise promise)` 方法，是 Netty 为 UDP 设计的。

2. NioSocketChannel

通过 `NioSocketChannel#disconnect()` 方法，应用程序里可以主动关闭 `NioSocketChannel` 通道。代码如下：

文章目录

- 1. 概述
- 2. NioSocketChannel
- 3. DefaultChannelPipeline
- 4. TailContext
- 666. 彩蛋

```
...ect() {  
...ct();
```

...Channel 抽象类，所以 `#disconnect()` 方法实际是 `AbstractChannel` 实现的。

- 在方法内部，会调用对应的 `ChannelPipeline#disconnect()` 方法，将 `disconnect` 事件在 `pipeline` 上传播。

3. DefaultChannelPipeline

`DefaultChannelPipeline#disconnect()` 方法，代码如下：

```
@Override
public final ChannelPipeline disconnect() {
    tail.disconnect();
    return this;
}
```

- 在方法内部，会调用 `TailContext#disconnect()` 方法，将 flush 事件在 pipeline 中，从尾节点向头节点传播。详细解析，见 [\[4. TailContext\]](#)。

4. TailContext

TailContext 对 `#flush()` 方法的实现，是从 `AbstractChannelHandlerContext` 抽象类继承，代码如下：

```
@Override
public ChannelFuture disconnect() {
    return disconnect(newPromise());
}

@Override
public ChannelFuture disconnect(final ChannelPromise promise) {
    // 判断是否为合法的 Promise 对象
    if (isValidPromise(promise, false)) {
        // cancelled
        return promise;
    }

    final AbstractChannelHandlerContext next = findContextOutbound();
    EventExecutor executor = next.executor();
    if (executor.inEventLoop()) {
        // <1> 如果没有 disconnect 操作，则执行 close 事件在 pipeline 上
        // Translate disconnect to close if the channel has no notion of disconnect-reconnect.
        // So far, UDP/IP is the only transport that has such behavior.
```

文章目录

1. 概述
2. NioSocketChannel
3. DefaultChannelPipeline
4. TailContext
666. 彩蛋

```
        metadata().hasDisconnect()) {
            next.invokeClose(promise);
        } else {
            // 如果没有 disconnect 操作，则执行 disconnect 事件在 pipeline 上
            next.invokeDisconnect(promise);
        }
    }
}, promise, null);
}
```

```
    return promise;
}
```

- 在 <1> 处，调用 ChannelMetadata#hasDisconnect() 方法，判断 Channel 是否支持 disconnect 操作。
 - 如果支持，则转换执行 close 事件在 pipeline 上。后续的逻辑，就是《精尽 Netty 源码解析 —— Channel (七) 之 close 操作》。
 - 如果不支持，则保持执行 disconnect 事件在 pipeline 上。
- 支持 disconnect 操作的 Netty Channel 实现类有：

```
private static final ChannelMetadata METADATA = new ChannelMetadata(true);           NioDatagramChannel.java 65
private static final ChannelMetadata METADATA = new ChannelMetadata(true);           OioDatagramChannel.java 62
private static final ChannelMetadata METADATA_DISCONNECT = new ChannelMetadata(true); EmbeddedChannel.java 60
private static final ChannelMetadata METADATA = new ChannelMetadata(true);           EpollDatagramChannel.java 50
private static final ChannelMetadata METADATA = new ChannelMetadata(true);           KQueueDatagramChannel.java 50
```

- 和文头，我们提到的，只有 Java 原生 UDP DatagramSocket 支持是一致的。从 So far, UDP/IP is the only transport that has such behavior. 的英文注释，也能证实这一点。
- 不支持 disconnect 操作的 Netty Channel 实现类有：

```
private static final ChannelMetadata METADATA = new ChannelMetadata(false, 16);      AbstractServerChannel.java 32
private static final ChannelMetadata METADATA = new ChannelMetadata(false, 16);      AbstractNioByteChannel.java 45
private static final ChannelMetadata METADATA = new ChannelMetadata(false);          AbstractOioByteChannel.java 40
private static final ChannelMetadata METADATA = new ChannelMetadata(false);          LocalChannel.java 56
private static final ChannelMetadata METADATA = new ChannelMetadata(false, 16);      NioServerSocketChannel.java 46
private static final ChannelMetadata METADATA = new ChannelMetadata(false, 1);       OioServerSocketChannel.java 48
private static final ChannelMetadata METADATA_NO_DISCONNECT = new ChannelMetadata(false); EmbeddedChannel.java 59
private static final ChannelMetadata METADATA = new ChannelMetadata(false);          FailedChannel.java 29
```

- 和文头，我们提到的，只有 Java 原生 NIO SocketChannel 不支持是一致的。

因为本系列，暂时不分享 UDP 相关的内容，所以对“执行 disconnect 事件在 pipeline 上”就不解析了。

666. 彩蛋

水更一篇，本来以为 Netty NIO Channel 的 disconnect 操作是个骚操作。

文章目录

总访问量 次

- 1. 概述
- 2. NioSocketChannel
- 3. DefaultChannelPipeline
- 4. TailContext
- 666. 彩蛋