Netty 异步事件驱动NIO 与原生NIO对比

原生NIO 服务端 code

public class NIOServer {

//通道管理器

private Selector selector;

public void initServer(int port) throws IOException {

// 获得一个ServerSocket通道

ServerSocketChannel serverChannel = ServerSocketChannel.open();

// 设置通道为非阻塞

serverChannel.configureBlocking(false);

// 将该通道对应的ServerSocket绑定到port端口

serverChannel.socket().bind(new InetSocketAddress(port));

// 获得一个通道管理器

this.selector = Selector.open();

//将通道管理器和该通道绑定，并为该通道注册SelectionKey.OP\_ACCEPT事件,注册该事件后，

//当该事件到达时，selector.select()会返回，如果该事件没到达selector.select()会一直阻塞。

serverChannel.register(selector, SelectionKey.OP\_ACCEPT);

}

public static void main(String[] args) throws IOException {

NIOServer server = new NIOServer();

server.initServer(8000);

}

}