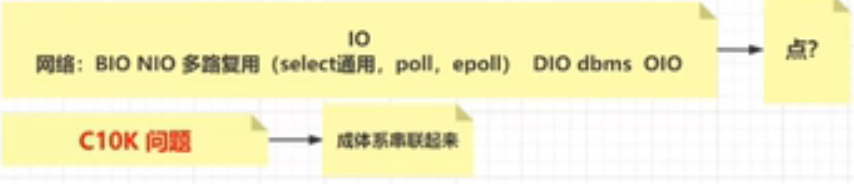
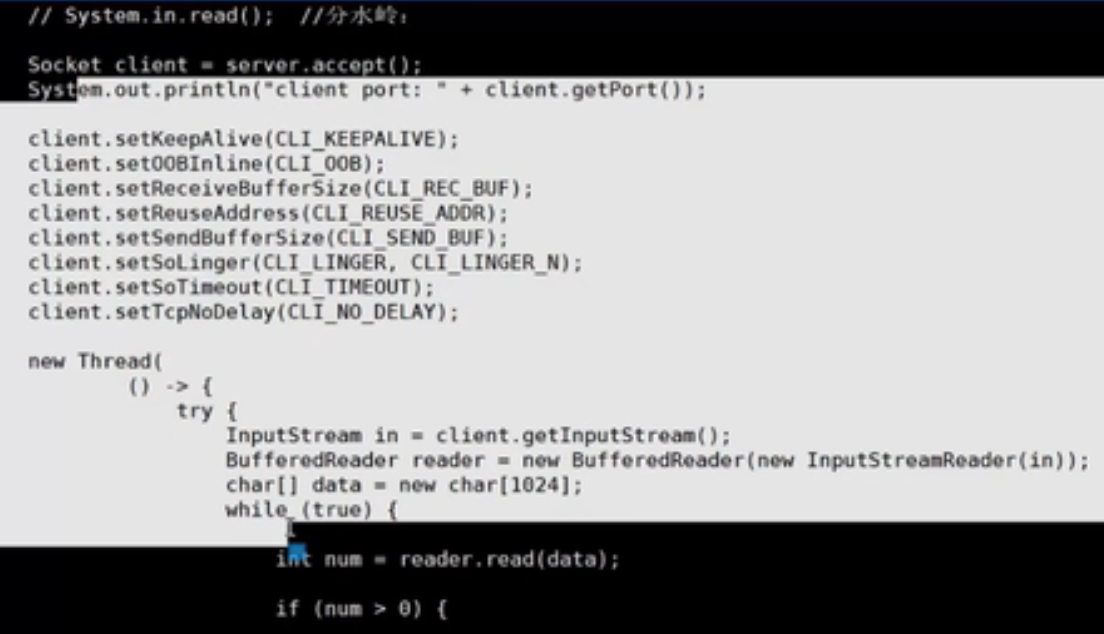
一、演示



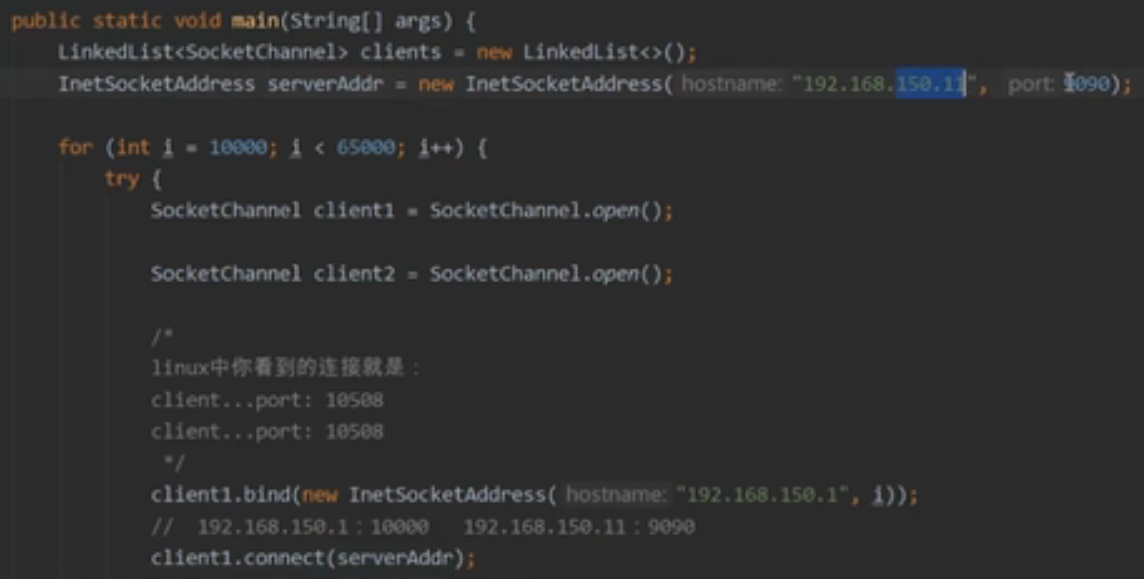
1. BIO 每连接，每线程

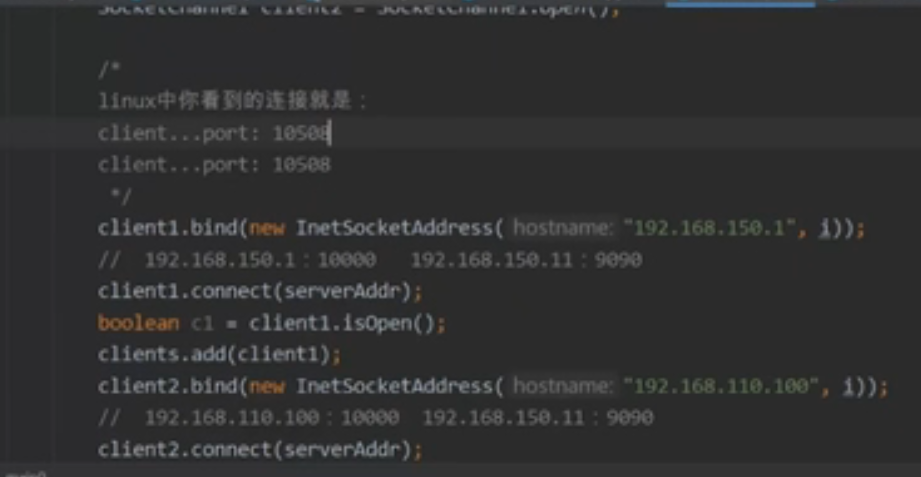
服务器：（阻塞，抛新线程）

阻塞原因：等待客户端发数据... => 所以需要抛线程



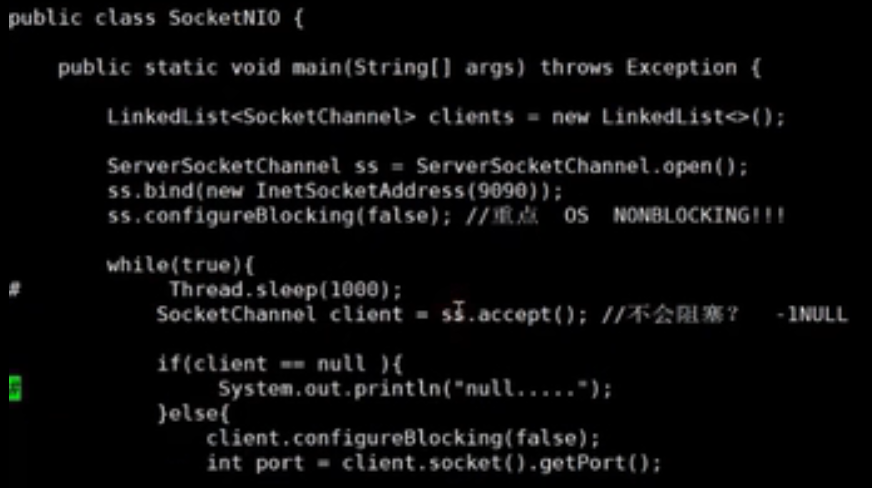
客户端：

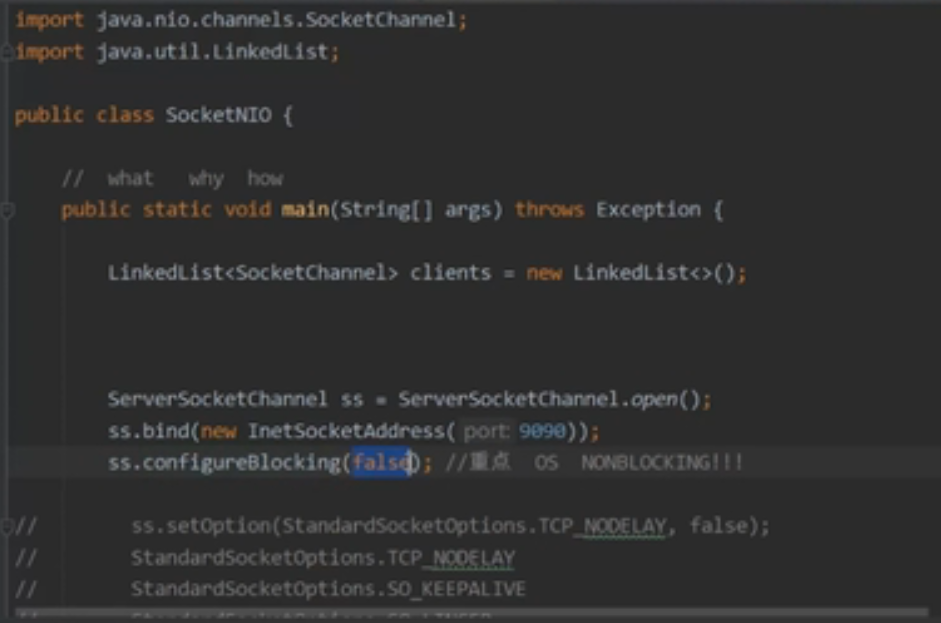




1. NIO

服务器（不阻塞，不抛新线程）

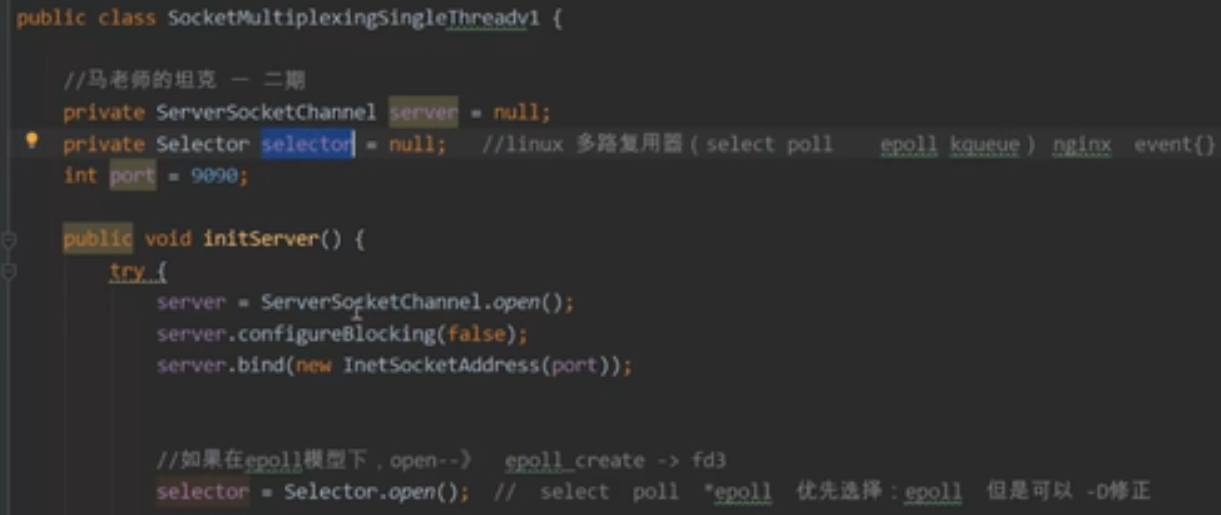


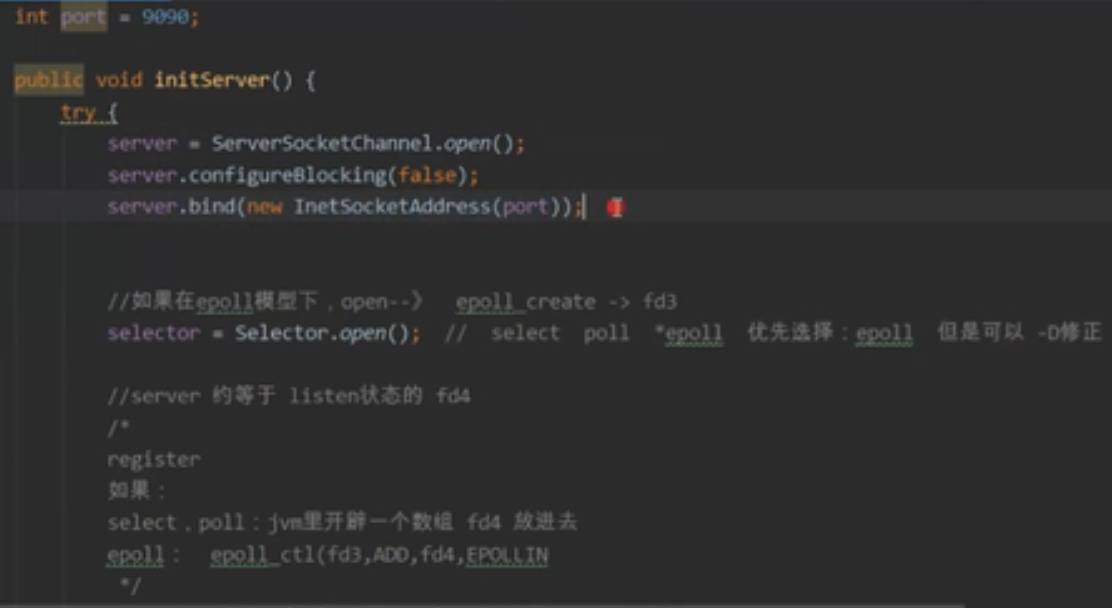


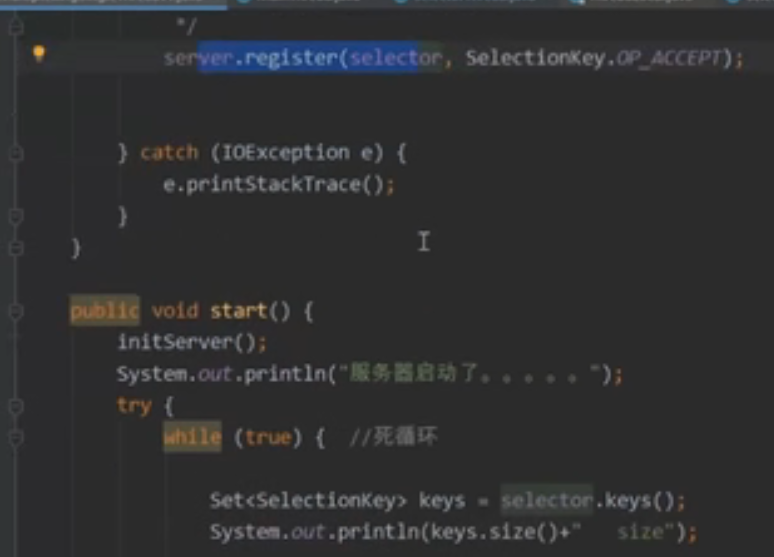
客户端：同上

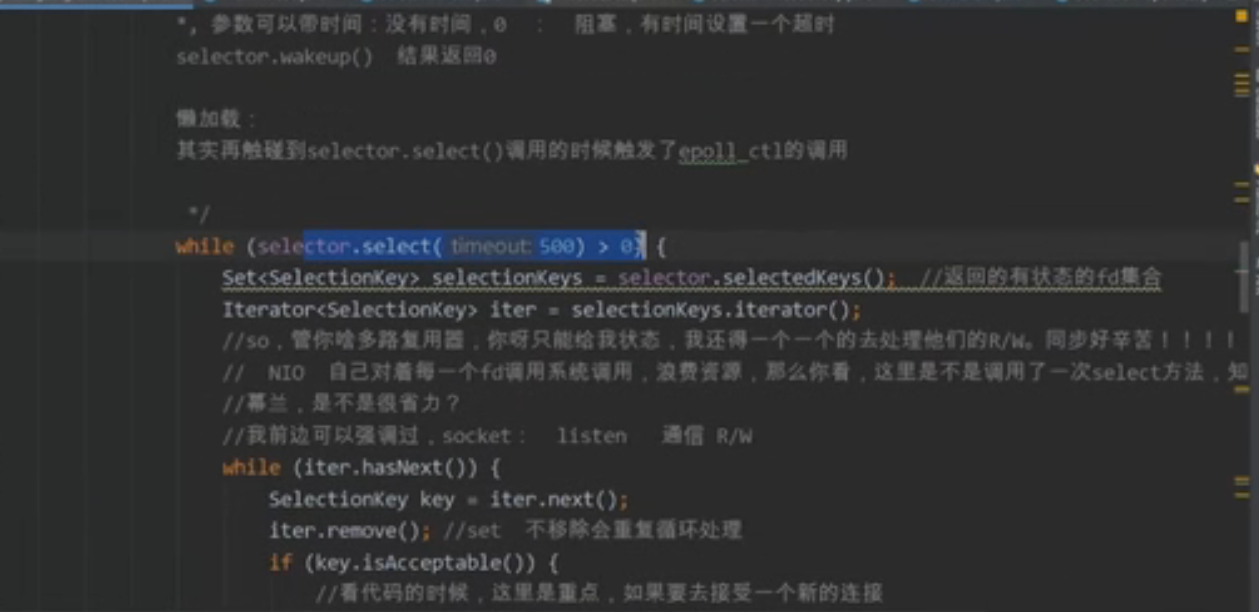
1. 多路复用（比NIO快了非常多）

服务器：

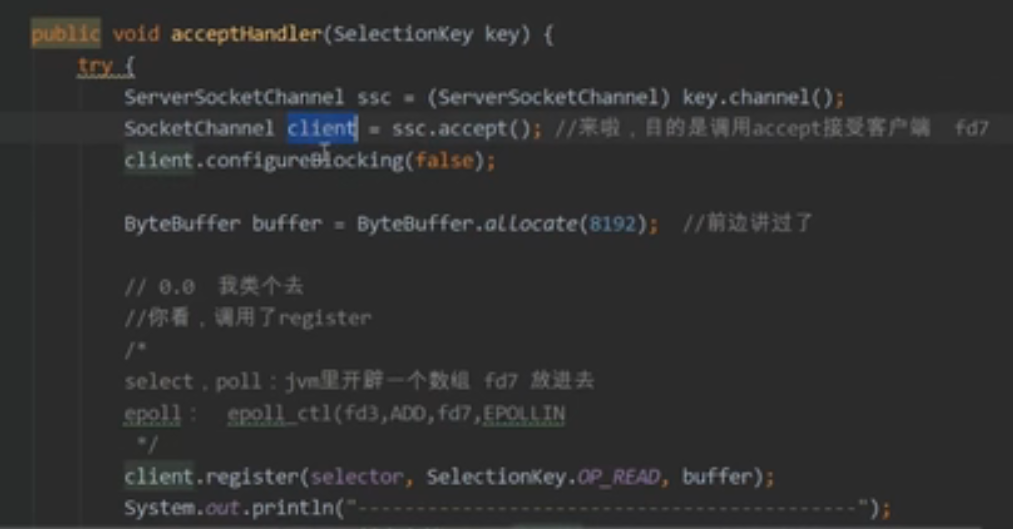


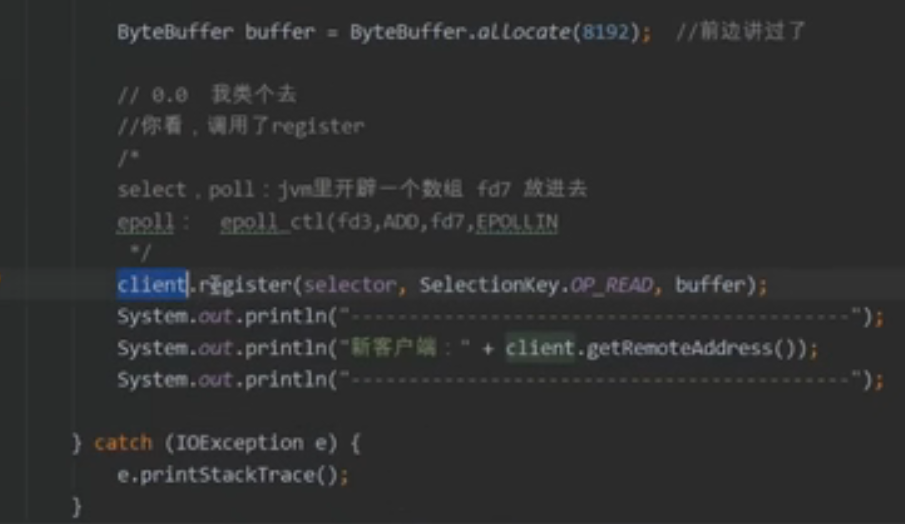






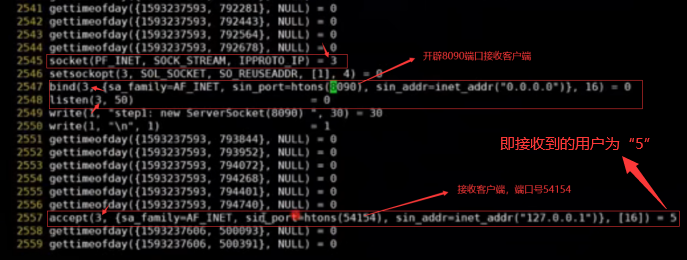




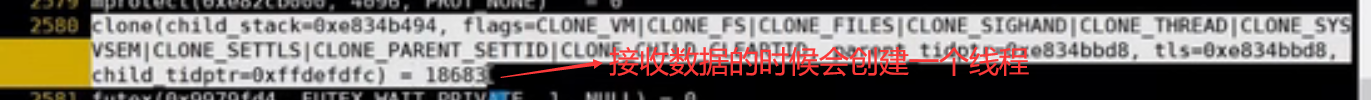


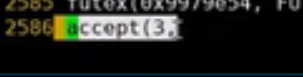
客户端：同上

1. 服务器端和底层交互
2. BIO

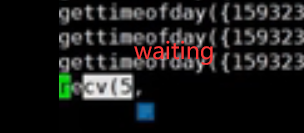


其中：

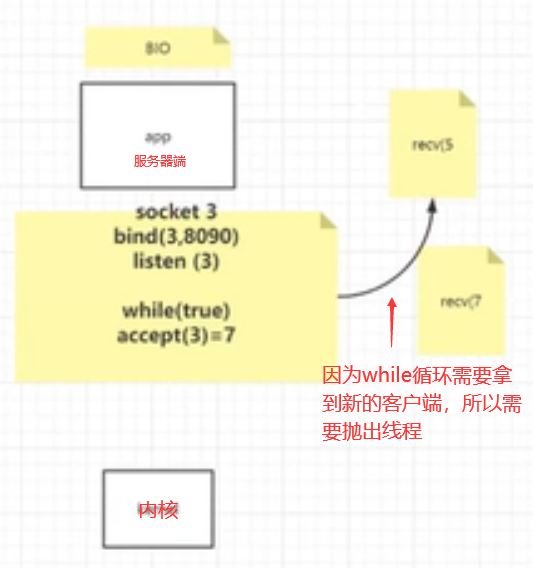


最后因为没有数据阻塞了

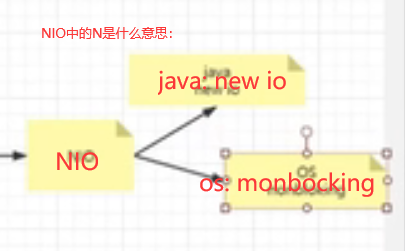
打开接收数据的18683：

 正在等待客户端发数据 --> 抛出

综上：

慢，资源消耗大，因为阻塞抛线程

1. NIO

（os: 操作系统）