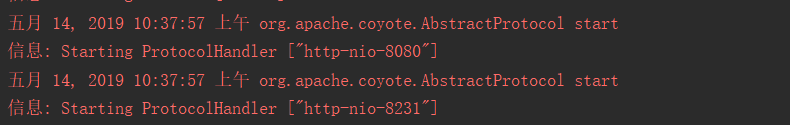
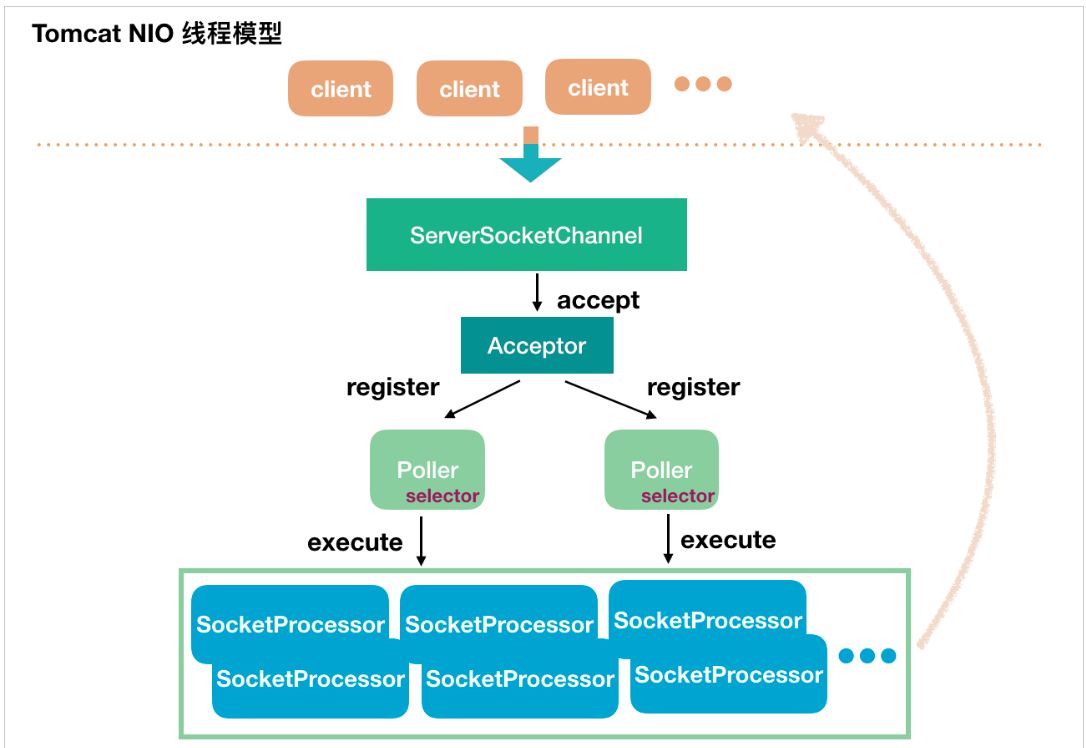
1. Tomcat启动示例，可以指定多个端口。一个端口对于一个connector。



1. Tomcat NIO线程模型。Acceptor(默认创建一个实例)连接器负责接收新网络连接，通过轮询策略register到某个Poller实例(创建一个或两个实例，如果是单核CPU则创建一个，封装了selector)，poller实例的selector.select()循环监听读写事件。当来了一个“感兴趣”的事件后分配给连接池去执行。



1. 默认最终执行读写事件的线程池大小：默认 corePoolSize 10, maxPoolSize 200。
2. Tomcat通过在acceptor中对闭锁(LimitLatch)的获取来控制总连接的数量(默认10000)，如果连接数量达到了最大的限制，那么将会被阻塞，直到有连接关闭为止。这样acceptor的线程就又被唤醒了。
3. Tomcat也支持AIO的方式，默认使用HTTP/1.1 NIO方式。
4. 可以发现很多开源框架都使用生命周期类来管理状态。Tomcat中使用LifecycleState(封装Lifecycle)枚举类来标记Acceptor的执行状态。