# 简介

星期四,2019年9月19日　　8:57

## 一、概述

1. NIO是java提供的一套用于传输数据的机制
2. BIO - BlockingIO - 同步阻塞式IO - File、UDP、TCP

   Server --- accept

   Client --- connect
3. NIO - NewIO - NonBlockingIO - 同步非阻塞式IO
4. NIO中的三大组件：Buffer、Channel、Selector

## 二、BIO的缺点

1. 一对一连接：每连接过来一个客户端，那么在服务器端就要产生一个线程取处理这个连接。那么就意味着如果过来大量的客户端，需要产生大量的线程，就会导致服务器卡顿甚至崩溃
2. 客户端连接之后如果不发生任何操作依然会占用服务器端的线程，也就意味着服务器端有大量的线程会被占用，并且这些线程的占用本质上并没有产生任何的效果

## 三、Buffer - 缓冲区

1. 在NIO中用于进行数据的**存储**
2. 底层是基于数组来进行存储的，只能存储基本类型的数据
3. 针对八种基本类型提供了7个子类，其中没有针对boolean类型的子类。因为数据的存储和传输的基本形式都是字节，所以主要掌握ByteBuffer
4. 重要位置
   a. capacity：容量位。用于标记当前缓冲区的容量/大小
   b. limit：限制位。用于限定position所能达到的最大下标。默认和容量位是重合的
   c. position：操作位。类似于数组中的下标，用于指向要操作的位置。默认是第0位
   d. mark：标记位。用于进行标记，通常是用于避免数据大批量产生错误。注意，标记位默认是不启用的

5. 重要操作
    a. flip：翻转缓冲区。将limit挪到position上，然后将position归零，标记位置为-1
    b. clear：清空缓冲区。position归零，limit挪到capacity上，mark置为-1
    c. reset：重置缓冲区。将position挪到mark上
    d. rewind：重绕缓冲区。将position归零，将mark置为-1

## 四、Channel - 通道

1. 在NIO中，用于进行数据的**传输**
2. Channel可以实现双向传输
3. Channel默认是阻塞的，可以手动设置非阻塞
4. 针对不同的场景提供了不同的子类：

    File：FileChannel

    UDP：DatagramChannel

    **TCP：SocketChannel、ServerSocketChannel**

## 五、Selector - 多路复用选择器

1. Selector针对事件来进行**选择**的
2. 利用Selector可以实现一对多的连接效果
3. Selector是面向通道进行选择，并且要求被选择的通道必须是非阻塞的