# 概念

## IO

IO指数据读写(内核空间和用户空间的相互拷贝)，一旦拿到数据后就不属于IO了。

## BIO

BIO(Blocking IO)，同步阻塞IO，用户线程被阻塞在数据读写上。

## NIO

JDK1.4引入NIO(Non-blocking IO)，非阻塞IO（操作线程非阻塞）。

NIO的读写函数可以立即返回，用户线程在数据读写时可以去做其他事情，这时是非阻塞的。但是数据读写阻塞在IO多路复用上，这个过程仍然是同步的，所以它是非阻塞同步IO。

## AIO（NIO2.0）

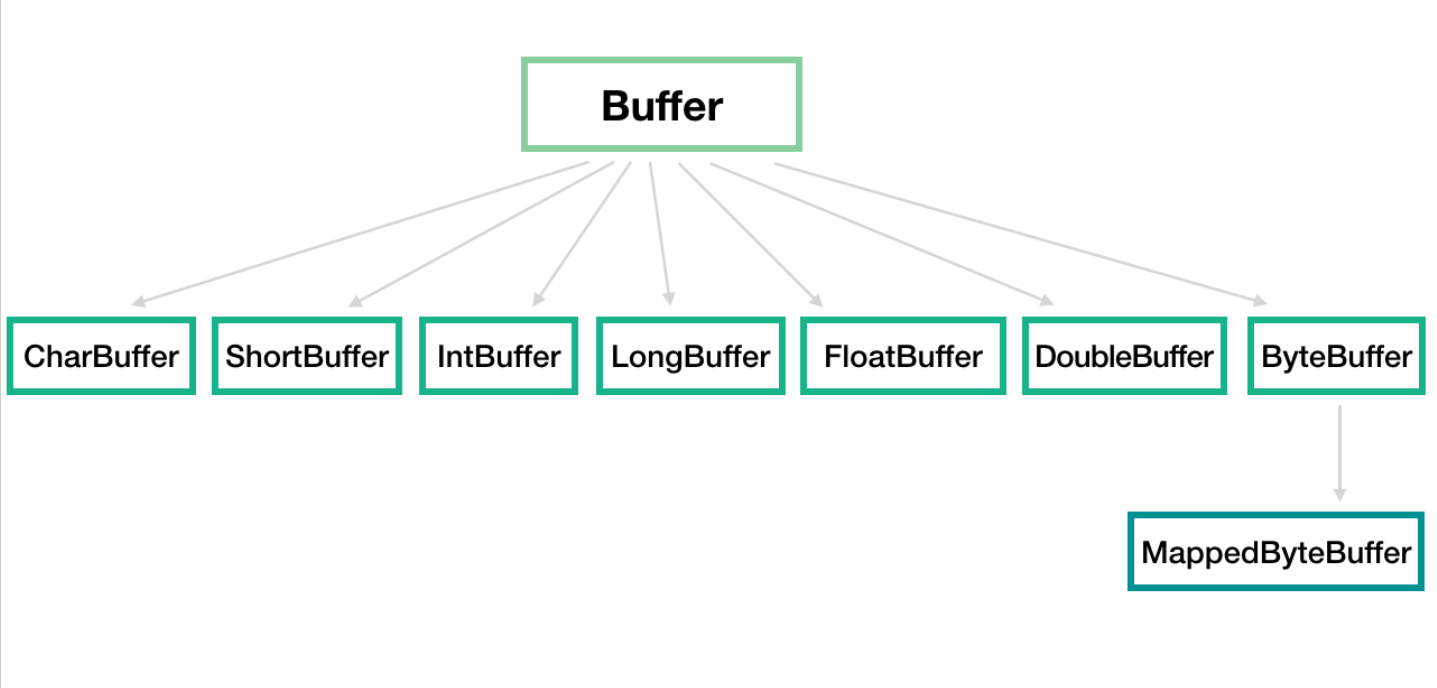
Jdk1.7引入NIO2.0==>AIO（Async Non-bloking IO）异步非阻塞IO，在NIO的基础上，数据读写均由操作系统内核完成，读写完成后会回调用户线程操作数据。

# NIO核心

NIO支持阻塞与非阻塞两种模式。 阻塞模式下, 实际上与流IO差不多；非阻塞模式下, Channels与Selector配合。

## Buffer

NIO用于读写数据的缓冲区。



### Capacity

缓冲区容量，一旦 Buffer 的容量达到 capacity，需要清空 Buffer，才能重新写入值。

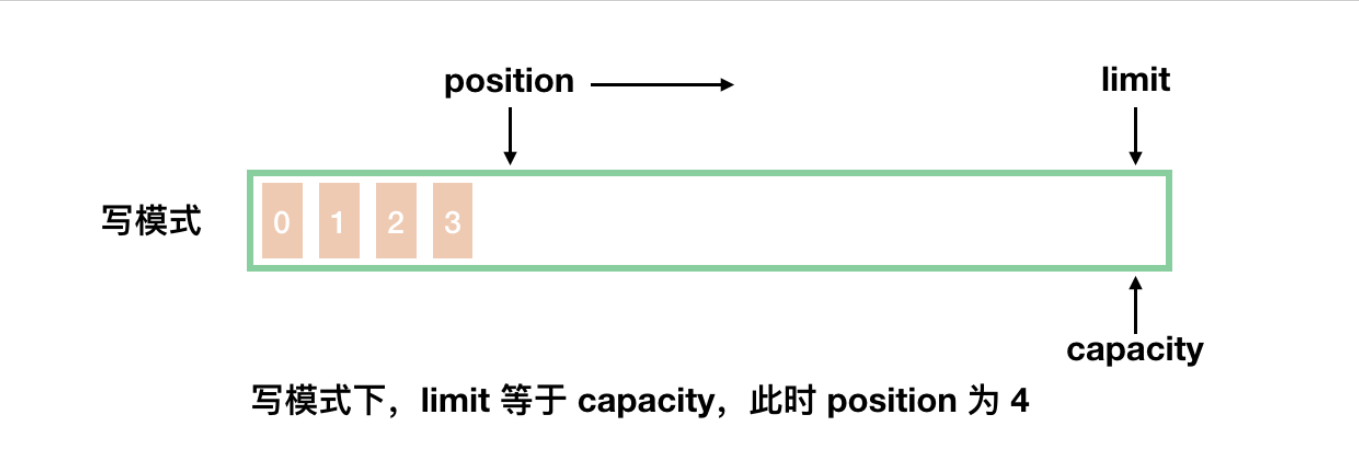
### Position

position 的初始值是 0，每往 Buffer 中写入一个值，position 就自动加 1，代表下一次的写入位置。读操作的时候也是类似的，每读一个值，position 就自动加 1。

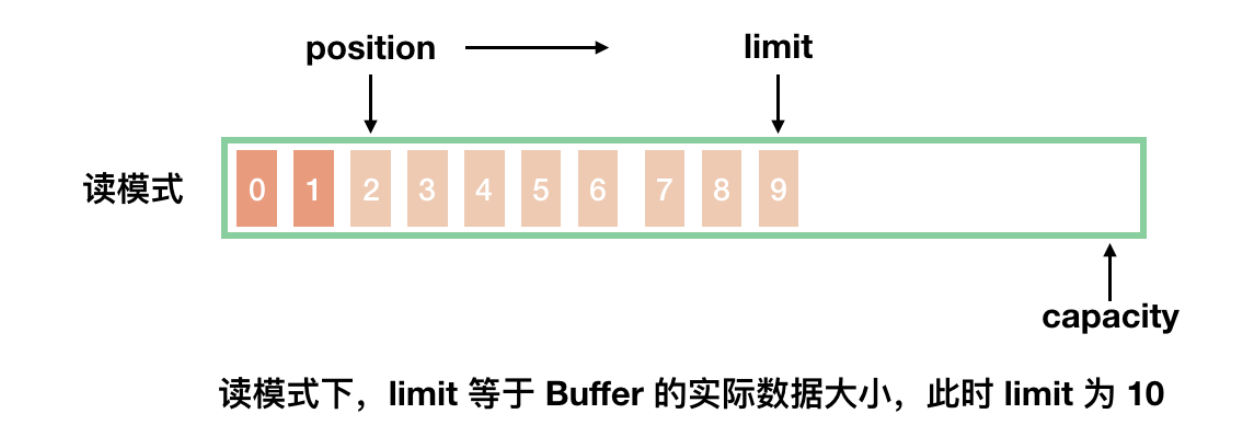
从写操作模式到读操作模式切换的时候（flip），position 会归零，这样就可以从头开始读写了。

### Limit

写模式，Limit表示最大能写入的大小，此时Limit与Capacity一致。

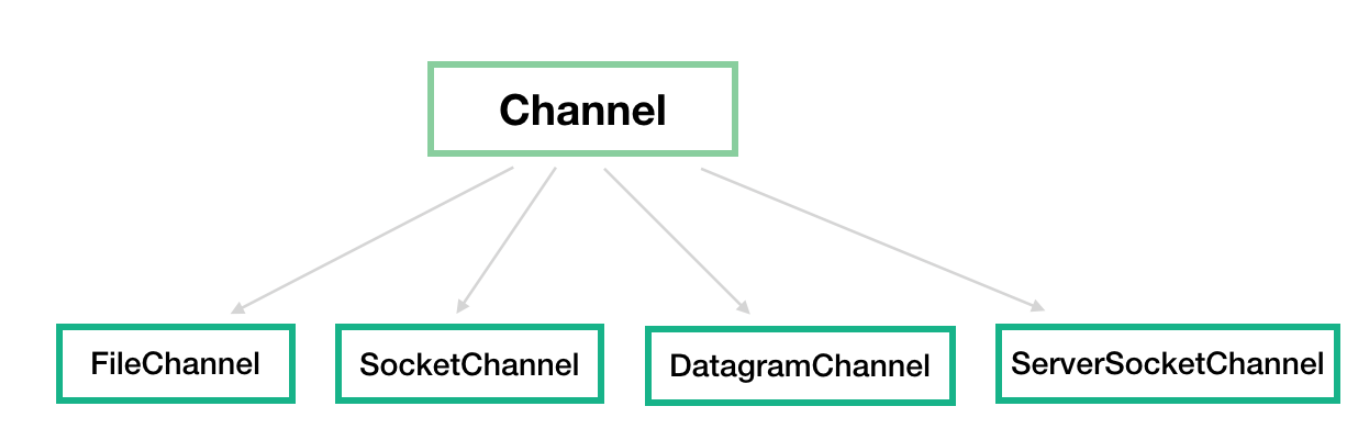


读模式，Limit表示Buffer的实际大小。



### 用法

## Channel



## Selector

# 相关

## IO多路复用

# 参考

<https://mp.weixin.qq.com/s/EVequWGVMWV5Ki2llFzdHg>

<https://www.javadoop.com/post/java-nio>