

Java认证辅导:非阻塞I/O简介Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_Java_E8_AE_A4_E8_AF_81_c104_644516.htm 在网络应用中，一般可以采用同步I/O（阻塞I/O）和非阻塞I/O两种方式进行数据通讯。这两种方式并非互相排斥和互相取代。我们可以在平时的应用中单独采用其中一种通讯方式，也可以混合使用这两种通讯方式。在本文中就什么是非阻塞I/O以及为什么要使用这种通讯方式进行了介绍，在下一篇文章中给出了一个简单的例子来演示在网络应用中如何使用非阻塞I/O进行通讯。

一、什么是非阻塞I/O 我们可以将同步I/O称为阻塞I/O，非阻塞I/O称为异步I/O.在本书中采用了比较常用的叫法：同步I/O和非阻塞I/O.虽然它们的叫法不同，但含义是一样的。读者在阅读其他书时应注意这一点。在讲解什么是非阻塞I/O之前，首先应了解什么是同步I/O，这里的同步指的是什么。同步这个概念在程序设计中主要是指代码按顺序执行的过程。如在Java程序中的main方法，如果不使用多线程，这个方法中的代码一定是从前往后按顺序执行的。这就叫做同步。如果使用了多线程，从宏观角度来看会有不同的代码段同时执行，这就叫做异步。在同步I/O中的同步概念也类似，也就是说，在I/O通讯过程中，只要是某一步的通讯未结束，就无法进行其他的通讯。那么这里的同步指的是什么呢？要回答这个问题之前，首先让我们先回忆一下，在网络通讯中有哪些地方可能会被阻塞。如果读者看了前面的章节就会知道答案。对于客户端来说，有两个地方可能会被阻塞：连接服务器（调用connect方法时）和读写数据。而在服务端也有两个地方可

能会被阻塞：等待客户端请求（调用accept方法时）和读写数据（在一般情况下，写数据不会被阻塞，但如果网络环境比较差的时候，客户端和服务端的写数据操作也可能发生阻塞现象）。也就是说，可以设置超时时间的地方就可能被阻塞。而同步 I/O 中的同步就是指除了以下两种情况外程序会一直处于等待状态：1. 连接服务器、读写数据或等待客户端请求正常地执行。2. 在等待超时时间后，抛出了超时异常。在上面我们了解了什么是同步 I/O. 而非阻塞 I/O 和同步 I/O 最明显的不同就是同步 I/O 所有可能被阻塞的地址在非阻塞 I/O 中都不会被阻塞。如在读取数据时，如果数据暂时无法被读取。那么在非阻塞 I/O 中会立刻返回，以便程序可以执行其他的代码，然后系统会不断侦测这个未完成的读取操作，直到可以继续读数据时再来完成这个操作。Java 在 JDK1.4 及以后版本中提供了一套 API 来专门操作非阻塞 I/O，我们可以在 java.nio 包及其子包中找到相关的类和接口。由于这套 API 是 JDK 新提供的 I/O API，因此，也叫 New I/O，这就是包名 nio 的由来。这套 API 由三个主要的部分组成：缓冲区（Buffers）、通道（Channels）和非阻塞 I/O 的核心类组成。这三部分的详细内容将在本章的后面介绍。

二、为什么要使用非阻塞 I/O

在使用同步 I/O 的网络应用中，如果要同时处理多个客户端请求，或是在客户端要同时和多个服务器进行通讯，就必须使用多线程来处理。也就是说，将每一个客户端请求分配给一个线程来单独处理。这样做虽然可以达到我们的要求，但同时又会带来另外一个问题。由于每创建一个线程，就要为这个线程分配一定的内存空间（也叫工作存储器），而且操作系统本身也对线程的总数有一定的限制。如果客户端的请求过多

，服务端程序可能会因为不堪重负而拒绝客户端的请求，甚至服务器可能会因此而瘫痪。当然，可以使用线程池（将在第三部分讲解）来缓解服务器的压力，但这并不能解决客户端因访问过于密集而造成的服务器拒绝响应的问题。虽然在服务端还有请求缓冲区作为保障，但这个缓冲区的大小是有限的（一般为50），如果客户端的请求数远超过这个数，客户端还是会收到拒绝服务的信息。在这种情况下，使用非阻塞I/O就可以解决这个问题。由于使用非阻塞I/O的程序一般是单线程的（有时可能将使用非阻塞I/O的程序段放到一个单独的线程里，而主线程负责处理用户的输入），因此，服务端接收的客户端请求数并不随着工作线程数的增加而增加。所以使用非阻塞I/O模式就不会受到操作系统对线程总数的限制，也不会占用大量的服务器资源。非阻塞I/O虽然可以到达在处理大量客户端请求的同时，又不占用大量的服务器资源的目的。但这种通讯方式并不能完全取代同步I/O.如非阻塞I/O并不适合象FTP服务器那样需要保持连接状态的应用（原因将在以后的章节中说明）。非阻塞I/O一般应用在服务端比较多一些，因为客户端一般并不需要处理大量的连接（但某些应用除外，如象百度、Google的Web Spider，需要同时下载多个网页，这时就需要在客户端建立大量的连接来满足需求），而服务端程序一般需要接收并处理大量的客户端请求，因此，就需要使用多线程（使用同步I/O）或非阻塞I/O来达到这个目的。如果某个服务端应用处理的客户端请求没那么多时，使用多线程和同步I/O可能会更好一点，因为这种方式要比非阻塞I/O方式更灵活。在前面一直将非阻塞I/O和网络应用放到一起讲。其实非阻塞I/O并不等于网络。我们也可

以将非阻塞I/O应用到非网络的应用中，如文件复制。由于同步I/O是基于字节流的，而非阻塞I/O是基于缓冲区和通道的。因此，从理论上，所操作的文件越大，非阻塞I/O的优势越能体现出来。而对于比较小的文件操作，这两种方式的效率差不多。根据实验得知，复制一个4G左右的文件，一般情况下，非阻塞I/O方式比同步I/O方式快大约15%左右。编辑特别推荐: Java核心API需要掌握的程度 Java编程实例：Java版农历和阳历转换源码 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com