

ImportNew

- [首页](#)
- [所有文章](#)
- [资讯](#)
- [Web](#)
- [架构](#)
- [基础技术](#)
- [书籍](#)
- [教程](#)
- [Java小组](#)
- [工具资源](#)

- 导航条 - ▼

Java NIO系列教程（7）：FileChannel

2016/04/03 | 分类：[教程](#) | [0 条评论](#) | 标签：[Java NIO](#), [selector](#)

分享到：

17 译文出处：[周泰](#) 原文出处：[Jakob Jenkov](#)

Java NIO中的FileChannel是一个连接到文件的通道。可以通过文件通道读写文件。

FileChannel无法设置为非阻塞模式，它总是运行在阻塞模式下。

打开FileChannel

在使用FileChannel之前，必须先打开它。但是，我们无法直接打开一个FileChannel，需要通过使用一个InputStream、OutputStream或RandomAccessFile来获取一个FileChannel实例。下面是通过RandomAccessFile打开FileChannel的示例：

```
1 RandomAccessFile aFile = new RandomAccessFile("data/nio-data.txt", "rw");
2 FileChannel inChannel = aFile.getChannel();
```

从FileChannel读取数据

调用多个read()方法之一从FileChannel中读取数据。如：

```
1 ByteBuffer buf = ByteBuffer.allocate(48);
2 int bytesRead = inChannel.read(buf);
```

首先，分配一个Buffer。从FileChannel中读取的数据将被读到Buffer中。

然后，调用FileChannel.read()方法。该方法将数据从FileChannel读取到Buffer中。read()方法返回的int值表示了有多少字节被读到了Buffer中。如果返回-1，表示到了文件末尾。

向FileChannel写数据

使用FileChannel.write()方法向FileChannel写数据，该方法的参数是一个Buffer。如：

```
1 String newData = "New String to write to file..." + System.currentTimeMillis();
2
3 ByteBuffer buf = ByteBuffer.allocate(48);
4 buf.clear();
5 buf.put(newData.getBytes());
6
7 buf.flip();
8
9 while(buf.hasRemaining()) {
10     channel.write(buf);
11 }
```

注意FileChannel.write()是在while循环中调用的。因为无法保证write()方法一次能向FileChannel写入多少字节，因此需要重复调用write()方法，直到Buffer中已经没有尚未写入通道的字节。

关闭FileChannel

用完FileChannel后必须将其关闭。如：

```
1 channel.close();
```

FileChannel的position方法

有时可能需要在FileChannel的某个特定位置进行数据的读/写操作。可以通过调用position()方法获取FileChannel的当前位置。

也可以通过调用position(long pos)方法设置FileChannel的当前位置。

这里有两个例子：

```
1 long pos = channel.position();
2 channel.position(pos + 123);
```



如果将位置设置在文件结束符之后，然后试图从文件通道中读取数据，读方法将返回-1 —— 文件结束标志。

如果将位置设置在文件结束符之后，然后向通道中写数据，文件将撑大到当前位置并写入数据。这可能导致“文件空洞”，磁盘上物理文件中写入的数据间有空隙。

FileChannel的size方法

FileChannel实例的size()方法将返回该实例所关联文件的大小。如：

```
1 long fileSize = channel.size();
```

FileChannel的truncate方法

可以使用FileChannel.truncate()方法截取一个文件。截取文件时，文件中指定长度后面的部分将被删除。如：

```
1 channel.truncate(1024);
```

这个例子截取文件的前1024个字节。

FileChannel的force方法

FileChannel.force()方法将通道里尚未写入磁盘的数据强制写到磁盘上。出于性能方面的考虑，操作系统会将数据缓存在内存中，所以无法保证写入到FileChannel里的数据一定会即时写到磁盘上。要保证这一点，需要调用force()方法。

force()方法有一个boolean类型的参数，指明是否同时将文件元数据（权限信息等）写到磁盘上。

下面的例子同时将文件数据和元数据强制写到磁盘上：

```
1 | channel.force(true);
```

本系列：

- [Java NIO系列教程（1）：Java NIO 概述](#)
- [Java NIO系列教程（2）：Channel](#)
- [Java NIO系列教程（3）：Buffer](#)
- [Java NIO系列教程（4）：Scatter/Gather](#)
- [Java NIO系列教程（5）：通道之间的数据传输](#)
- [Java NIO系列教程（6）：Selector](#)
- [Java NIO系列教程（7）：FileChannel](#)

17



相关文章

- [Java NIO系列教程（6）：Selector](#)
- [Selector 实现原理](#)
- [epoll 浅析以及 nio 中的 Selector](#)
- [攻破JAVA NIO技术壁垒](#)
- [Java NIO系列教程（12）：Java NIO与IO](#)
- [Java NIO系列教程（11）：Pipe](#)
- [Java NIO系列教程（10）：Java NIO DatagramChannel](#)
- [Java NIO系列教程（9）：ServerSocketChannel](#)
- [Java NIO系列教程（8）：SocketChannel](#)
- [Java NIO系列教程（5）：通道之间的数据传输](#)

发表评论

Comment form

Name*

姓名

邮箱*

请填写邮箱

网站 (请以 http://开头)

评论内容*

(*) 表示必填项

[提交评论](#)

还没有评论。

[« Java NIO系列教程（6）：Selector](#)
[Java NIO系列教程（8）：SocketChannel »](#)

Search for:



- [本周热门文章](#)
- [本月热门](#)
- [热门标签](#)

[0 记一次集群内无可用 http 服务问题...](#)[1 Java 技术之垃圾回收机制](#)[2 公司编程竞赛之最长路径问题](#)[3 Java 中的十个"单行代码编程" \(O...](#)[4 Java 中 9 个处理 Exception ...](#)

- 5 [HttpClient 以及 Json 传递的...](#)
- 6 [浅析 Spring 中的事件驱动机制](#)
- 7 [浅析分布式下的事件驱动机制 \(PubS...](#)
- 8 [探索各种随机函数 \(Java 环境...](#)
- 9 [Java 守护线程概述](#)



最新评论

- 
Re: [攻破JAVA NIO技术壁垒](#)
Hi，请到伯乐在线的小组发帖提问，支持微信登录。链接是： <http://group.jobbole....> 唐尤华
- 
Re: [攻破JAVA NIO技术壁垒](#)
TCP服务端的NIO写法 服务端怎么发送呢。原谅小白 菜鸟 
- 
Re: [关于 Java 中的 double check ...](#)
volatile 可以避免指令重排啊。所以double check还是可以用的。 hipilee
- 
Re: [Spring4 + Spring MVC + M...](#)
Hi，请到伯乐在线的小组发帖提问，支持微信登录。链接是： <http://group.jobbole....> 唐尤华
- 
Re: [Spring4 + Spring MVC + M...](#)
我一直不太明白，spring的bean容器和springmvc的bean容器之间的关系。 hw_绝影
- 
Re: [Spirng+SpringMVC+Maven+Myba...](#)
很好，按照步骤，已经成功。 莫凡
- 
Re: [Spring中@Transactional事务...](#)
声明式事务可以用aop来实现,分别是jdk代理和cglib代理,基于接口和普通类.在同一个类中一个方... chengjiliang
- 
Re: [关于 Java 中的 double check ...](#)

在JDK1.5之后，用volatile关键字修饰_INSTANCE属性 就能避免因指令重排导致的对象...
Byron

关于ImportNew

ImportNew 专注于 Java 技术分享。于2012年11月11日 11:11正式上线。是的，这是一个很特别的时刻：)

ImportNew 由两个 Java 关键字 import 和 new 组成，意指：Java 开发者学习新知识的网站。import 可认为是学习和吸收，new 则可认为是新知识、新技术圈子和新朋友.....



联系我们

Email : ImportNew.com@gmail.com

新浪微博 : [@ImportNew](#)

推荐微信号



ImportNew



安卓应用频道



Linux爱好者



反馈建议 : ImportNew.com@gmail.com

广告与商务合作QQ : 2302462408

推荐关注

[小组](#) – 好的话题、有启发的回复、值得信赖的圈子

[头条](#) – 写了文章？看干货？去头条！

[相亲](#) – 为IT单身男女服务的征婚传播平台

[资源](#) – 优秀的工具资源导航

[翻译](#) – 活跃 & 专业的翻译小组

[博客](#) – 国内外的精选博客文章

[设计](#) – UI,网页，交互和用户体验

[前端](#) – JavaScript, HTML5, CSS

[安卓](#) – 专注Android技术分享

[iOS](#) – 专注iOS技术分享

[Java](#) – 专注Java技术分享

[Python](#) – 专注Python技术分享

© 2017 ImportNew