

Phinecos(洞庭散人)

专注于开源技术的研究与应用

Java NIO非阻塞服务器示例

以前一直用的是“ervery thread per connection”的服务器端模式，今天试了下NIO非阻塞模式的服务器。 不过java不能实现I/O完成端口模型，这点很遗憾

```
package com.vista.Server;

import java.io.IOException;
import java.net.InetSocketAddress;
import java.net.ServerSocket;
import java.nio.ByteBuffer;
import java.nio.channels.SelectionKey;
import java.nio.channels.Selector;
import java.nio.channels.ServerSocketChannel;
import java.nio.channels.SocketChannel;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Set;

public class SelectorServer
{
    private static int DEFAULT_SERVERPORT = 6018; // 默认端口
    private static int DEFAULT_BUFFERSIZE = 1024; // 默认缓冲区大小为1024字节

    private ServerSocketChannel channel;
    private LinkedList<SocketChannel> clients;
    private Selector readSelector;
    private ByteBuffer buffer; // 字节缓冲区
    private int port;

    public SelectorServer(int port) throws IOException
    {
        this.port = port;
        this.clients = new LinkedList<SocketChannel>();
        this.channel = null;
        this.readSelector = Selector.open(); // 打开选择器
    }
}
```

导航

博客园

首页

新随笔

联系

订阅 XML

管理

统计

随笔 - 592

文章 - 2

评论 - 1238

引用 - 117

公告

+ Google™

+ 订阅到 抓虾

+ 订阅到 鲜果

+ 订阅到 有道

+ 订阅到 QQ邮箱

姓名：邓以克

网名：phinecos(洞庭散人)

MSN: phinecos@msn.com

我的主页 个人资料

我的闪存 发短消息

搜索

随笔分类(726)

.Net(33) (rss)

ACM(69) (rss)

Android(9) (rss)

Assembly(4) (rss)

C/C++/VC++(245) (rss)

COM/ATL/ActiveX(16) (rss)

Flex(9) (rss)

```

        this.buffer = ByteBuffer.allocate(DEFAULT_BUFFERSIZE);
    }
    // 服务器程序在服务循环中调用serviceClients()方法为已接受的客户服务
    public void serviceClients() throws IOException
    {
        Set keys;
        Iterator it;
        SelectionKey key;
        SocketChannel client;
        // 在readSelector上调用select()方法, 参数1代表如果调用select的时候
        那么阻塞最多1秒钟等待可用的客户端连接
        if(readSelector.select(1) > 0)
        {
            keys = readSelector.selectedKeys(); // 取得代表端通道的键集

            it = keys.iterator();
            // 遍历, 为每一个客户服务
            while(it.hasNext())
            {
                key = (SelectionKey)it.next();
                if(key.isReadable())
                { // 如果通道可读, 那么读此通道到buffer中
                    int bytes;
                    client = (SocketChannel)key.channel(); // 取得键对应
                    的通道
                    buffer.clear(); // 清空缓冲区中的内容, 设置好position, limit, 准备接受数据
                    bytes = client.read(buffer); // 从通道中读数据到缓冲中
                    , 返回读取得字节数
                    if(bytes >= 0)
                    {
                        buffer.flip(); // 准备将缓冲中的数据写回到通道中
                        client.write(buffer); // 数据写回到通道中
                    }
                    else if(bytes < 0)
                    { // 如果返回小于零的值代表读到了流的末尾
                        clients.remove(client);
                        // 通道关闭时, 选择键也被取消
                        client.close();
                    }
                }
            }
        }
    }
}

```

[Java\(115\) \(rss\)](#)
[linux\(27\) \(rss\)](#)
[Mozilla扩展\(21\) \(rss\)](#)
[Others\(30\) \(rss\)](#)
[PHP\(1\) \(rss\)](#)
[python \(rss\)](#)
[Search Engine\(16\) \(rss\)](#)
[程序人生\(29\) \(rss\)](#)
[计算机图形学\(71\) \(rss\)](#)
[考研心得\(31\) \(rss\)](#)

[随笔档案\(592\)](#)

[2010年4月 \(4\)](#)
[2010年3月 \(3\)](#)
[2009年11月 \(1\)](#)
[2009年10月 \(5\)](#)
[2009年9月 \(18\)](#)
[2009年8月 \(9\)](#)
[2009年6月 \(7\)](#)
[2009年5月 \(16\)](#)
[2009年3月 \(9\)](#)
[2009年2月 \(5\)](#)
[2009年1月 \(6\)](#)
[2008年12月 \(17\)](#)
[2008年11月 \(57\)](#)
[2008年10月 \(23\)](#)
[2008年9月 \(15\)](#)
[2008年8月 \(13\)](#)
[2008年7月 \(33\)](#)
[2008年6月 \(28\)](#)
[2008年5月 \(23\)](#)
[2008年4月 \(16\)](#)
[2008年3月 \(14\)](#)
[2008年2月 \(2\)](#)
[2008年1月 \(13\)](#)
[2007年12月 \(14\)](#)

```

    }

    public void registerClient(SocketChannel client) throws IOException
    { // 配置和注册代表客户连接的通道对象
        client.configureBlocking(false); // 设置此通道使用非阻塞模式

        client.register(readSelector, SelectionKey.OP_READ); // 将这个通道注册到选择器上
        clients.add(client); // 保存这个通道对象
    }

    public void listen() throws IOException
    { // 服务器开始监听端口, 提供服务
        ServerSocket socket;
        SocketChannel client;
        channel = ServerSocketChannel.open(); // 打开通道
        socket = channel.socket(); // 得到与通道相关的socket对象
        socket.bind(new InetSocketAddress(port), 10); // 将socket绑定在制定的端口上
        // 配置通道使用非阻塞模式, 在非阻塞模式下, 可以编写多道程序同时避免使用复杂的多线程

        channel.configureBlocking(false);
        try
        {
            while(true)
            { // 与通常的程序不同, 这里使用channel.accept() 接受客户端连接请求, 而不是在socket对象上调用accept(), 这里在调用accept() 方法时如果通道配置为非阻塞模式, 那么accept() 方法立即返回null, 并不阻塞

                client = channel.accept();
                if(client != null)
                {
                    registerClient(client); // 注册客户信息
                }
                serviceClients(); // 为以连接的客户服务
            }
        }
        finally
        {
            socket.close(); // 关闭socket, 关闭socket会同时关闭与此socket关联的通道
        }
    }

    public static void main(String[] args) throws IOException

```

2007年11月 (24)

2007年10月 (13)

2007年9月 (13)

2007年8月 (28)

2007年7月 (17)

2007年6月 (12)

2007年5月 (7)

2007年4月 (19)

2007年3月 (11)

2007年1月 (1)

2006年12月 (3)

2006年11月 (6)

2006年10月 (19)

2006年9月 (11)

2006年8月 (23)

2006年7月 (4)

2006年6月 (26)

2006年5月 (4)

相册

my picture

常去的站点

cnBeta

codeproject

developerWorks

sourceforge

我的好友

susu

vandalor

涛哥

我的站点

记录我的生活

我的百度空间

我的优盘

我在CSDN的blog

```

    {
        System.out.println("服务器启动...");
        SelectorServer server = new SelectorServer(SelectorServer.DE
FAULT_SERVERPORT);
        server.listen(); //服务器开始监听端口，提供服务
    }
}

```

修改版本:

```

package com.vista.Server;

import java.io.BufferedWriter;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.InetSocketAddress;
import java.net.ServerSocket;
import java.nio.ByteBuffer;
import java.nio.CharBuffer;
import java.nio.channels.FileChannel;
import java.nio.channels.SelectionKey;
import java.nio.channels.Selector;
import java.nio.channels.ServerSocketChannel;
import java.nio.channels.SocketChannel;
import java.nio.charset.Charset;
import java.nio.charset.CharsetDecoder;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Set;

public class SelectorServer
{
    private static int DEFAULT_SERVERPORT = 6018; //默认端口
    private static int DEFAULT_BUFFERSIZE = 1024; //默认缓冲区大小为1024
字节

```

积分与排名

积分 - 713518

排名 - 47

最新评论 XML

1. Re: 《基于MFC的OpenGL编程》 Part 19
Creating a Virtual Reality Walkthrough
Application

您好! 看了您的文章很好, 我最近也在学OPENGL, 您有空可以发一份《基于MFC的OPENGL编程》, 给我吗? 我的邮箱是lu9573@yahoo.cn, 谢谢了

--陆名

2. Re: 《基于MFC的OpenGL编程》 Part 17
Shadows

高手, 我这里用的是VS2005的MFC基于对话框的环境, 我把include和lib都配好了, 然后一加这行#include <PixieLib.h> 就出问题了 fatal error LNK...

--逍遥路飞

3. Re: 《基于MFC的OpenGL编程》 Part 7 Animation

真是受益匪浅, 希望您有时间能不能发送给我一份完整代码。ldjv2003@163.com

--不想注册

4. Re: Struts+Spring+Hibernate快速入门
恩, 有代码不错, 最好是有全局的思路就不错了。貌似有点在推springside的意思。

--CowboyRyan

5. Re: GStreamer学习笔记 (一)

恩, 写得很详细, 受教了, 能否再推荐些关于GStreamer编程方面的详细资料

--ajaxhe

阅读排行榜

1. Self Introduction(15184)

2. TinyXML: 一个优秀的C++ XML解析器(15161)

3. 基于MFC的OpenGL绘图(11028)

```

private static String DEFAULT_CHARSET = "GB2312";//默认码集
private static String DEFAULT_FILENAME = "bigfile.dat";
private ServerSocketChannel channel;
private LinkedList<SocketChannel> clients;
private Selector selector;//选择器
private ByteBuffer buffer;//字节缓冲区
private int port;
private Charset charset;//字符集
private CharsetDecoder decoder;//解码器

public SelectorServer(int port) throws IOException
{
    this.port = port;
    this.clients = new LinkedList<SocketChannel>();
    this.channel = null;
    this.selector = Selector.open();//打开选择器
    this.buffer = ByteBuffer.allocate(DEFAULT_BUFFERSIZE);
    this.charset = Charset.forName(DEFAULT_CHARSET);
    this.decoder = this.charset.newDecoder();
}

private class HandleClient
{
    private String strGreeting = "welcome to VistaQQ";
    public HandleClient() throws IOException
    {
    }
    public String readBlock()
    { //读块数据
        return this.strGreeting;
    }
    public void close()
    {
    }
}

protected void handleKey(SelectionKey key) throws IOException
{ //处理事件
    if (key.isAcceptable())
    { // 接收请求

```

4. IIS默认网站（停止）错误0x8ffe2740解决办法(10254)

5. ActiveMQ笔记（一）(9268)

评论排行榜

1. 基于朴素贝叶斯分类器的文本分类算法（下）(94)

2. 基于MFC的OpenGL绘图(71)

3. IIS默认网站（停止）错误0x8ffe2740解决办法(44)

4. 基于JMF RTP的音视频传输(40)

5. 浅谈基于Mozilla ThunderBird的扩展开发(37)

60天内阅读排行

1. 今年的计划(587)

2. 浅谈Java多线程的同步问题(307)

3. Java中利用package机制实现条件编译(249)

4. 浅谈ActionScript3中的原型对象(156)

5. Ubuntu下安装Fcitx中文输入法(133)

```

        ServerSocketChannel server = (ServerSocketChannel) key
        .channel(); //取出对应的服务器通道
        SocketChannel channel = server.accept();
        channel.configureBlocking(false);
        channel.register(selector, SelectionKey.OP_READ); //客户
socket通道注册读操作
    }
    else if (key.isReadable())
    { // 读信息
        SocketChannel channel = (SocketChannel) key.channel();
        int count = channel.read(this.buffer);
        if (count > 0)
        {
            this.buffer.flip();
            CharBuffer charBuffer = decoder.decode(this.buffer);
            System.out.println("Client >>" + charBuffer.toString
        );

            SelectionKey wKey = channel.register(selector,
                SelectionKey.OP_WRITE); //为客户socket通道注册写操作
            wKey.attach(new HandleClient());
        }
        else
        { //客户已经断开
            channel.close();
        }
        this.buffer.clear(); //清空缓冲区
    }
    else if (key.isWritable())
    { // 写事件
        SocketChannel channel = (SocketChannel) key.channel();
        HandleClient handle = (HandleClient) key.attachment();
        //取出处理者
        ByteBuffer block = ByteBuffer.wrap(handle.readBlock().
getBytes());
        channel.write(block);
        // channel.socket().getInputStream().(block);
        //
        PrintWriter out = new PrintWriter(new BufferedWriter(
new OutputStreamWriter(
        //
            channel.socket().getOutputStream()), true
        ));
        //
        out.write(block.toString());
    }
}

```

```

    }
    public void listen() throws IOException
    { //服务器开始监听端口, 提供服务
        ServerSocket socket;
        channel = ServerSocketChannel.open(); // 打开通道
        socket = channel.socket(); //得到与通道相关的socket对象
        socket.bind(new InetSocketAddress(port)); //将socket绑定在制
        定的端口上
        //配置通道使用非阻塞模式, 在非阻塞模式下, 可以编写多道程序同时避免使用复
        杂的多线程
        channel.configureBlocking(false);
        channel.register(selector, SelectionKey.OP_ACCEPT);
        try
        {
            while(true)
            { // 与通常的程序不同, 这里使用channel.accept()接受客户端连
              接请求, 而不是在socket对象上调用accept(), 这里在调用accept()方法时如果通道配置为
              非阻塞模式, 那么accept()方法立即返回null, 并不阻塞
                this.selector.select();
                Iterator iter = this.selector.selectedKeys().iterato
            r();
                while(iter.hasNext())
                {
                    SelectionKey key = (SelectionKey)iter.next();
                    iter.remove();
                    this.handleKey(key);
                }
            }
        }
        catch(IOException ex)
        {
            ex.printStackTrace();
        }
    }
    public static void main(String[] args) throws IOException
    {
        System.out.println("服务器启动...");
        SelectorServer server = new SelectorServer(SelectorServer.DE
        FAULT_SERVERPORT);
        server.listen(); //服务器开始监听端口, 提供服务
    }
}

```

```
L  
}
```

作者: 洞庭散人

出处: <http://phinecos.cnblogs.com/>

本文版权归作者和博客园共有, 欢迎转载, 但未经作者同意必须保留此段声明, 且在文章页面明显位置给出原文连接, 否则保留追究法律责任的权利。

0

0

(请您对文章做出评价)

« 上一篇: [C++ Exercises \(十四\) --解线性方程组](#)

» 下一篇: [C++ Exercises \(十五\) --排序算法的简单实现](#)

posted on 2008-07-17 19:35 [Phinecos](#)(洞庭散人) 阅读(3538) 评论(1) 编辑 收藏 所属分类: [Java](#)

评论

#1楼 2009-01-20 09:39 [王凯迪](#)[未注册用户]

我正在学习NIO,多谢博主 [回复](#) [引用](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#) 。

[博客园首页](#) [IT新闻](#) [闪存](#) [知识库](#) [招聘](#)

找优秀程序员, 就在博客园

IT新闻:

- [开源会成为SOA的最终归宿吗?](#)
- [2010年中国信息产业十大发展趋势](#)
- [Unix类似火箭巴蒂尔 x86替代RISC为时尚早](#)
- [Metro设计文档开放下载, 下周将有更多惊喜期待](#)
- [CN域名又快回来了](#)

每天10分钟, 轻松学英语

专题: [Android](#) [iPad](#) [jQuery](#) [Windows 7](#)



推荐职位:

数据加载中...

网站导航:

[博客园首页](#) [IT新闻](#) [个人主页](#) [闪存](#) [程序员招聘](#) [社区](#) [博问](#) [网摘](#)



[China-pub](#) 计算机图书网上专卖店! 6.5万品种2-8折!

[China-Pub](#) 计算机绝版图书按需印刷服务

相关搜索:

[Java](#)

在知识库中查看:

[Java NIO非阻塞服务器示例](#)

Powered by:

[博客园](#)

Copyright © Phinecos(洞庭散人)