

导航

博客园
首页
新随笔
联系
XML 订阅
管理

< 2009年11月 >						
日	一	二	三	四	五	六
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

公告

我的主页 个人资料
我的闪存 发短消息
搜索

常用链接

我的随笔
我的空间
我的短信
我的评论
更多链接

随笔分类(1)

励志(1)(rss)

随笔档案(2)

基于Socket的Java网络编程集粹

事实上网络编程简单的理解就是两台计算机相互通讯数据而已，对于程序员而言，去掌握一种编程接口并使用一种编程模型相对就会显得简单的多了，Java SDK提供一些相对简单的Api来完成这些工作。Socket就是其中之一，对于Java而言，这些Api存在与java.net 这个包里面，因此只要导入这个包就可以准备网络编程了。

网络编程的基本模型就是客户机到服务器模型，简单的说就是两个进程之间相互通讯，然后其中一个必须提供一个固定的位置，而另一个则只需要知道这个固定的位置。并去建立两者之间的联系，然后完成数据的通讯就可以了，这里提供固定位置的通常称为服务器，而建立联系的通常叫做客户端，基于这个简单的模型，就可以进入网络编程啦。

Java对这个模型的支持有很多种Api，而这里我只想介绍有关Socket的编程接口，对于Java而言已经简化了Socket的编程接口。首先我们来讨论有关提供固定位置的服务方是如何建立的。Java提供了ServerSocket来对其进行支持.事实上当你创建该类的一个实力对象并提供一个端口资源你就建立了一个固定位置可以让其他计算机来访问你，ServerSocket server=new ServerSocket(6789);这里稍微要注意的是端口的分配必须是唯一的。因为端口是为了唯一标识每台计算机唯一服务的，另外端口号是从0~65535之间的，前1024个端口已经被Tcp/Ip 作为保留端口，因此你所分配的端口只能是1024个之后的。好了，我们有了固定位置.现在所需要的就是一根连接线了.该连接线由客户方首先提出要求。因此Java同样提供了一个Socket对象来对其进行支持，只要客户方创建一个Socket的实例对象进行支持就可以了。Socket client=new Socket(InetAddress.getLocalHost(),5678);客户机必须知道有关服务器的IP地址，对于着一点Java也提供了一个相关的类InetAddress 该对象的实例必须通过它的静态方法来提供，它的静态方法主要提供了得到本机IP 和通过名字或IP直接得到InetAddress的方法。

上面的方法基本可以建立一条连线让两台计算机相互交流了，可是数据是如何传输的呢？事实上I/O操作总是和网络编程息息相关的。因为底层的网络是继续数据的，除非远程调用，处理问题的核心在执行上，否则数据的交互还是依赖于IO操作的，所以你也必须导入java.io这个包.java的IO操作也不复杂，它提供了针对于字节流和Unicode的读者和写者，然后也提供了一个缓冲用于数据的读写。

```
BufferedReader in=new BufferedReader(new InputStreamReader(server.getInputStream()));  
PrintWriter out=new PrintWriter(server.getOutputStream());
```

上面两句就是建立缓冲并把原始的字节流转变为Unicode可以操作，而原始的字节流来源于Socket的两个方法，getInputStream()和getOutputStream()方，分别用来得到输入和输出，那么现在有了基本的模型和基本的操作工具，我们可以做一个简单的Socket例程了。

2006年11月 (1)

2006年6月 (1)

文章分类(7)

C#(rss)

java(2)(rss)

XML(rss)

其他技巧(2)(rss)

数据库技术(3)(rss)

心情物语(rss)

文章档案(7)

2008年11月 (1)

2008年9月 (1)

2008年1月 (1)

2007年11月 (1)

2007年4月 (1)

2006年6月 (2)

最新随笔

1. Word2007鼠标失效 解决办法(转)

2. 常用两种复制表数据SQL语句

3. Win2003系统中,插入移动硬盘U盘,无盘符显示解决方法

4. 删除指定表的数据

5. FROM 指定在 DELETE、SELECT 和 UPDATE 语句中使用的表、视图、派生表和连接表。

6. 取出11-25条之间的数据

7. 一位老工程师给年轻人的“十大忠告”

8. 基于Socket的Java网络编程集粹

9. 如何才算掌握Java (J2SE篇)

积分与排名

积分 - 4497

排名 - 8485

最新评论 XML

服务方:

```
import java.io.*;
import java.net.*;

public class MyServer {
    public static void main(String[] args) throws IOException{
        ServerSocket server=new ServerSocket(5678);
        Socket client=server.accept();
        BufferedReader in=new BufferedReader(new InputStreamReader(client.getInputStream()));
        PrintWriter out=new PrintWriter(client.getOutputStream());

        while(true){
            String str=in.readLine();
            System.out.println(str);
            out.println("has receive....");
            out.flush();
            if(str.equals("end"))
                break;
        }
        client.close();
    }
}
```

这个程序的主要目的在于服务器不断接收客户机所写入的信息只到，客户机发送"End"字符串就退出程序，并且服务器也会做出"Receive"为回应，告知客户机已接收到消息。

客户机代码:

```
import java.net.*;
import java.io.*;

public class Client{
    static Socket server;

    public static void main(String[] args)throws Exception{
        server=new Socket(InetAddress.getLocalHost(),5678);
        BufferedReader in=new BufferedReader(new InputStreamReader(server.getInputStream()));
        PrintWriter out=new PrintWriter(server.getOutputStream());
        BufferedReader wt=new BufferedReader(new InputStreamReader(System.in));

        while(true){
            String str=wt.readLine();
```

阅读排行榜

1. 一位老工程师给年轻人的“十大忠告” (55)
2. 取出11-25条之间的数据(28)

评论排行榜

1. 取出11-25条之间的数据(0)
2. 一位老工程师给年轻人的“十大忠告” (0)

```
out.println(str);
out.flush();
if(str.equals("end")){
break;
}
System.out.println(in.readLine());
}
server.close();
}
}
```

客户机代码则是接受客户键盘输入,并把该信息输出,然后输出"End"用来做退出标识。

这个程序只是简单的两台计算机之间的通讯,如果是多个客户同时访问一个服务器呢?你可以试着再运行一个客户端,结果是会抛出异常的。那么多个客户端如何实现呢?

其实,简单的分析一下,就可以看出客户和服务通讯的主要通道就是Socket本身,而服务器通过accept方法就是同意和客户建立通讯.这样当客户建立Socket的同时。服务器也会使用这一根连线来先后通讯,那么既然如此只要我们存在多条连线就可以了。那么我们的程序可以变为如下:

服务器:

```
import java.io.*;
import java.net.*;

public class MyServer {
public static void main(String[] args) throws IOException{
ServerSocket server=new ServerSocket(5678);
while(true){
Socket client=server.accept();
BufferedReader in=new BufferedReader(new InputStreamReader(client.getInputStream()));
PrintWriter out=new PrintWriter(client.getOutputStream());
while(true){
String str=in.readLine();
System.out.println(str);
out.println("has receive....");
out.flush();
if(str.equals("end"))
break;
}
}
```

```
client.close();
}
}
}
```

这里仅仅只是加了一个外层的**While**循环，这个循环的目的就是当一个客户进来就为它分配一个**Socket**直到这个客户完成一次和服务器的交互，这里也就是接受到客户的"**End**"消息.那么现在就实现了多客户之间的交互了。但是.问题又来了，这样做虽然解决了多客户,可是是排队执行的。也就是说当一个客户和服务器的完成一次通讯之后下一个客户才可以进来和服务器的交互，无法做到同时服务，那么要如何才能同时达到既能相互之间交流又能同时交流呢？很显然这是一个并行执行的问题了。所以线程是最好的解决方案。

那么下面的问题是如何使用线程.首先要做的事情是创建线程并使得其可以和网络连线取得联系。然后由线程来执行刚才的操作，要创建线程要么直接继承**Thread**要么实现**Runnable**接口，要建立和**Socket**的联系只要传递引用就可以了.而要执行线程就必须重写**run**方法，而**run**方法所做的事情就是刚才单线程版本**main**所做的事情，因此我们的程序变成了这样:

```
import java.net.*;
import java.io.*;

public class MultiUser extends Thread{
    private Socket client;

    public MultiUser(Socket c){
        this.client=c;
    }

    public void run(){
        try{
            BufferedReader in=new BufferedReader(new InputStreamReader(client.getInputStream()));
            PrintWriter out=new PrintWriter(client.getOutputStream());
            //Mutil User but can parallel
            while(true){
                String str=in.readLine();
                System.out.println(str);
                out.println("has receive....");
                out.flush();
                if(str.equals("end"))
                    break;
            }
            client.close();
        }
    }
}
```

```
}catch(IOException ex){
}finally{
}
}

public static void main(String[] args)throws IOException{
    ServerSocket server=new ServerSocket(5678);
    while(true){
        //transfer location change Single User or Multi User
        MultiUser mu=new MultiUser(server.accept());
        mu.start();
    }
}
}
```

我的类直接从**Thread**类继承了下来.并且通过构造函数传递引用和客户**Socket**建立了联系，这样每个线程就有了。一个通讯管道.同样我们可以填写run方法，把之前的操作交给线程来完成，这样多客户并行的**Socket**就建立起来了。

以上的代码使用的是

```
BufferedReader in=new BufferedReader(new InputStreamReader(client.getInputStream()));
PrintWriter out=new PrintWriter(client.getOutputStream());
```

还有一种方法是使用

```
DataInputStream isFromClient = new DataInputStream(client.getInputStream());
DataOutputStream osToClient = new DataOutputStream(client.getOutputStream());
```

关于这两种输入输出流的不同，我也只知道前一种对字符串支持比较好，后面对于读取一个字符串需要处理，但是可以支持很多种类型的输出。对于传递字符串而言前一种应该是很好的选择了

0 0

(请您对文章做出评价)

» 下一篇: [一位老工程师给年轻人的“十大忠告”](#)

posted on 2006-06-29 19:23 无忧浪子 阅读(41) 评论(0) 编辑 收藏 网摘 所属分类: java

昵称:

主页:

评论内容:



[注销](#) [订阅](#) [回复](#)

[使用**Ctrl+Enter**键快速提交]

IT新闻:

- 中国作家协会发维权通告: 要求谷歌尽快赔偿
- 谁能成为中国的亚马逊Kindle?
- 模仿文件夹U盘病毒攻击WIN7
- 英特尔高调进军国内动漫产业 提供软硬件支持
- **Google**被证实将推自有品牌手机 明年年初或上市

招聘信息:

- 方正软件（苏州）有限公司招聘**JAVA**软件工程师（开发部）
- 第一视频招聘项目经理
- 成都册盟高新科技发展有限公司招聘.**Net**架构工程师
- 招聘项目经理
- **360buy**京东商城招聘架构师
- 深圳市东方传奇网络科技有限公司招聘.**NET**程序员
- 亚旭电子科技（江苏）有限公司招聘.**NET**工程师

微软/劳动部双认证免费.NET培训

博客园个人主页上线测试

Windows 7专题

NHibernate专题



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

链接: 切换模板

导航: 网站首页 个人主页 社区 IT新闻 博问 闪存 网摘 招聘 找找看 Google搜索

相关搜索:

java

Powered by:

博客园

Copyright © 无忧浪子