

## § 13.3.3 ServerSocket对象与服务器端套接字\_1

- 服务器必须建立一个ServerSocket对象，该对象通过将客户端的套接字对象和服务端的一个套接字对象连接起来，从而达到连接的目的。
- 建立ServerSocket对象：

```
try{ ServerSocket serverForClient =new ServerSocket(2010);  
}  
catch(IOException e) {}
```

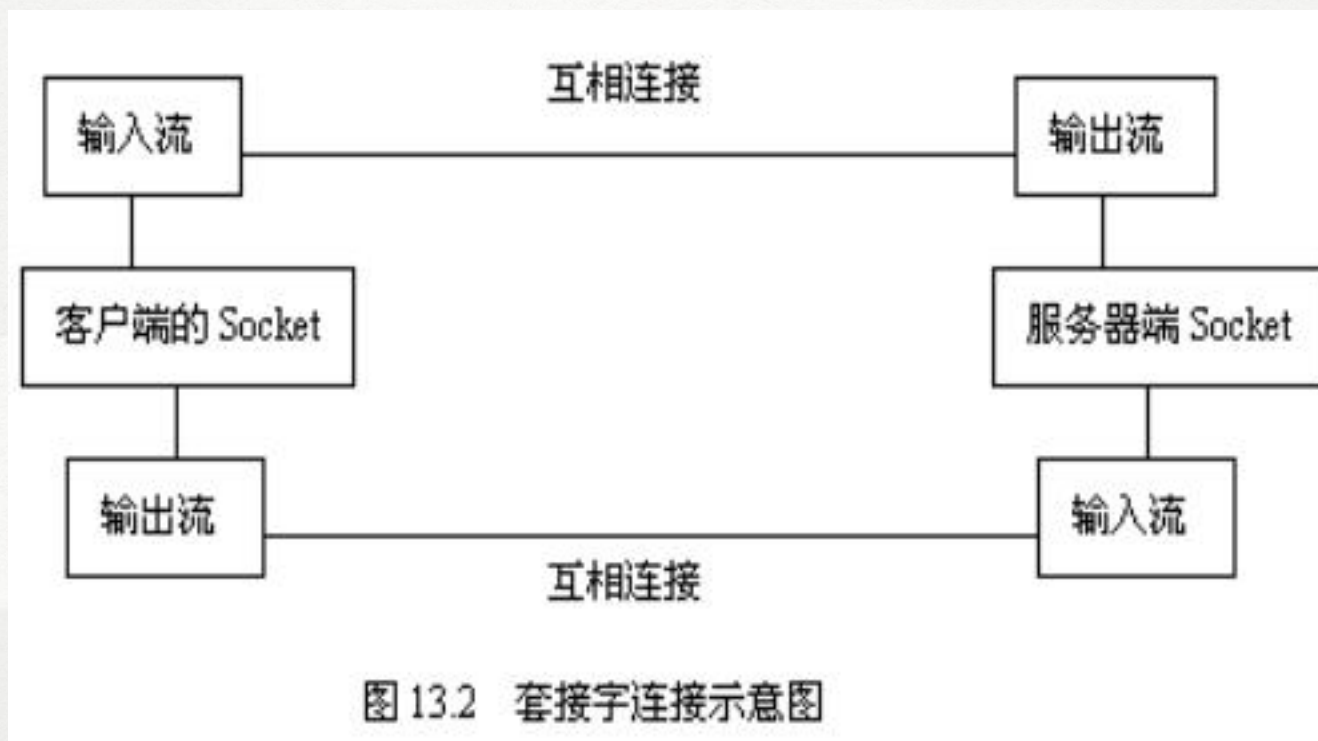
- 使用方法accept()将客户的套接字和服务端端的套接字连接起来，代码如下所示：

```
try{ Socket sc= serverForClient .accept();  
}  
catch(IOException e) {}
```

- 所谓“接收”客户的套接字连接就是accept()方法会返回一个和客户端Socket对象相连接的Socket对象。

## § 13.3.3 ServerSocket对象与服务器端套接字\_2

- 客户端的套接字获得的输入\输出流和服务器端的套接字获得的输出\输入流互相连接.





## § 13.3.3 ServerSocket对象与服务器端套接字\_2

- 例子3中，客户端向服务器问了三句话，服务器都给出了一一的回答。首先将例子3中服务器端的Server.java编译通过，并运行起来，等待客户的呼叫，然后运行客户端程序。客户端运行效果如图13.3，服务器端运行效果如图13.4。

客户端

(Client.java)

```
C:\client>java Client  
客户收到服务器的回答:南非  
客户收到服务器的回答:进入世界杯了  
客户收到服务器的回答:哈哈...问题真逗!
```

图 13.3 客户端

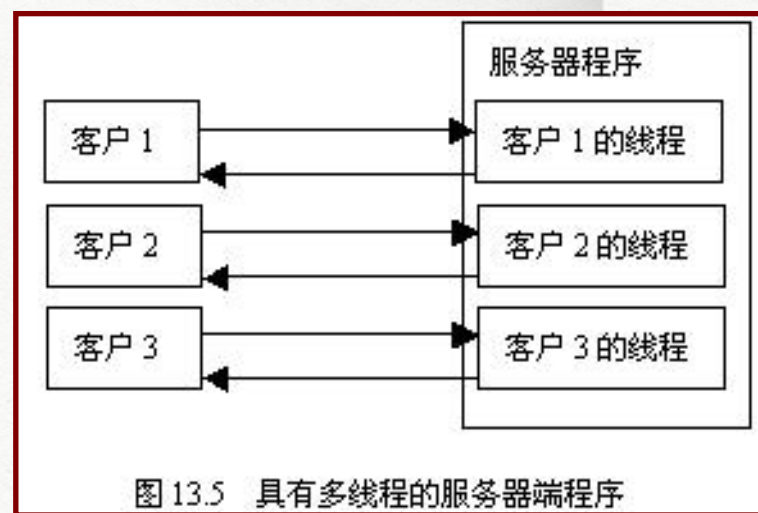
服务器端 (Server.java)

```
D:\Server>java Server  
等待客户呼叫  
服务器收到客户的提问:2010世界杯在哪举行?  
服务器收到客户的提问:巴西进入世界杯了码?  
服务器收到客户的提问:中国进入世界杯了码?
```

图 13.4 服务器端

## § 13.3.4 使用多线程技术

- 为了防止堵塞线程，服务器端收到一个客户的套接字后，就应该启动一个专门为该客户服务的线程。
- 在下面的例子4中，客户使用Socket类不带参数的构造方法Socket() 创建一个套接字对象，该对象需调用
  - `public void connect(SocketAddress endpoint) throws IOException`
- 请求和参数SocketAddress指定地址的套接字建立连接。
- 为了使用connect方法，可以使用SocketAddress的子类InetSocketAddress创建一个对象，InetSocketAddress的构造方法是：
  - `InetSocketAddress(InetAddress addr, int port)`





## § 13.3.4 使用多线程技术

- 例子4中，客户输入圆的半径并发送给服务器，服务器把计算出的圆的面积返回给客户。因此可以将计算量大的工作放在服务器端，客户负责计算量小的工作，实现客户-服务器交互计算，来完成某项任务。首先将例子4中服务器端的程序编译通过，并运行起来，等待客户的呼叫。客户端运行效果如图13.6，服务器端运行效果如图13.7。

客户端 (Client.java ,  
Read.java )

```
C:\client>java Client
输入服务器的IP:127.0.0.1
输入端口号:2010
输入圆的半径 (放弃请输入N):18
圆的面积:1017.8760197630929
输入圆的半径 (放弃请输入N):
```

图 13.6 客户端

服务器端(Server.java )

```
等待客户呼叫
客户的地址:/127.0.0.1
正在监听
等待客户呼叫
客户离开
客户的地址:/127.0.0.1
正在监听
等待客户呼叫
```

图 13.7 服务器端