

用Java实现多线程服务器程序 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022__E7_94_A8Java_E5_AE_9E_c104_145383.htm

摘要：在Java出现之前，编写多线程程序是一件烦琐且伴随许多不安全因素的事情。利用Java，编写安全高效的多线程程序变得简单，而且利用多线程和Java的网络包我们可以方便的实现多线程服务器程序。

---- Java是伴随Internet的大潮产生的，对网络及多线程具有内在的支持，具有网络时代编程语言的一切特点。从Java的当前应用看，Java主要用于在Internet或局域网上的网络编程，而且将Java作为主流的网络编程语言的趋势愈来愈明显。 ---- 一

、Java中的服务器程序与多线程 ---- 在Java之前，没有一种主流编程语言能够提供对高级网络编程的固有支持。在其他语言环境中，实现网络程序往往需要深入依赖于操作平台的网络API的技术中去，而Java提供了对网络支持的无平台相关性的完整软件包，使程序员没有必要为系统网络支持的细节而烦恼。 ---- Java软件包内在支持的网络协议为TCP/IP，也是当今最流行的广域网/局域网协议。Java有关网络的类及接口定义在java.net包中。客户端软件通常使用java.net包中的核心类Socket与服务器的某个端口建立连接，而服务器程序不同于客户机，它需要初始化一个端口进行监听，遇到连接呼叫，才与相应的客户机建立连接。Java.net包的ServerSocket类包含了编写服务器系统所需的一切。下面给出ServerSocket类的部分定义。

```
public class ServerSocket {
    public ServerSocket(int port) throws IOException ;
    public Socket accept() throws IOException ;
    public InetAddress getInetAddress() ;
    public int
```

getLocalPort() ; public void close() throws IOException ; public synchronized void setSoTimeout(int timeout) throws SocketException ; public synchronized int getSoTimeout() throws IOException ; }---- ServerSocket构造器是服务器程序运行的基础，它将参数port指定的端口初始化作为该服务器的端口，监听客户机连接请求。Port的范围是0到65536，但0到1023是标准Internet协议保留端口，而且在Unix主机上，这些端口只有root用户可以使用。一般自定义的端口号在8000到16000之间。仅初始化了ServerSocket还是远远不够的，它没有同客户机交互的套接字（Socket），因此需要调用该类的accept方法接受客户呼叫。Accept()方法直到有连接请求才返回通信套接字(Socket)的实例。通过这个实例的输入、输出流，服务器可以接收用户指令，并将相应结果回应客户机。ServerSocket类的getInetAddress和getLocalPort方法可得到该服务器的IP地址和端口。---- Java的多线程可谓是Java编程的精华之一，运用得当可以极大地改善程序的响应时间，提高程序的并行性。在服务器程序中，由于往往要接收不同客户机的同时请求或命令，因此可以对每个客户机的请求生成一个命令处理线程，同时对各用户的指令作出反应。在一些较复杂的系统中，我们还可以为每个数据库查询指令生成单独的线程，并行对数据库进行操作。实践证明，采用多线程设计可以很好的改善系统的响应，并保证用户指令执行的独立性。由于Java本身是"线程安全"的，因此有一条编程原则是能够独立在一个线程中完成的操作就应该开辟一个新的线程。 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