# 实验二 网络基础编程实验(JAVA\Python3)

**一、实验目的：**  
　　通过本实验，学习采用Socket（套接字）设计简单的网络数据收发程序，理解应用数据包是如何通过传输层进行传送的。   
二、**实验内容：**

Socket（套接字）是一种抽象层，应用程序通过它来发送和接收数据，就像应用程序打开一个文件句柄，将数据读写到稳定的存储器上一样。一个socket允许应用程序添加到网络中，并与处于同一个网络中的其他应用程序进行通信。一台计算机上的应用程序向socket写入的信息能够被另一台计算机上的另一个应用程序读取，反之亦然。

不同类型的socket与不同类型的底层协议族以及同一协议族中的不同协议栈相关联。现在TCP/IP协议族中的主要socket类型为流套接字（sockets sockets）和数据报套接字（datagram sockets）。流套接字将TCP作为其端对端协议（底层使用IP协议），提供了一个可信赖的字节流服务。一个TCP/IP流套接字代表了TCP连接的一端。数据报套接字使用UDP协议（底层同样使用IP协议），提供了一个"尽力而为"（best-effort）的数据报服务，应用程序可以通过它发送最长65500字节的个人信息。一个TCP/IP套接字由一个互联网地址，一个端对端协议（TCP或UDP协议）以及一个端口号唯一确定。

**2.1、采用TCP进行数据发送的简单程序（java/python3.5）**



****

**2.2、采用UDP进行数据发送的简单程序（java/python3.5）**



****

**2.3多线程\线程池对比（java/python3.5）**

当一个客户端向一个已经被其他客户端占用的服务器发送连接请求时，虽然其在连接建立后即可向服务器端发送数据，服务器端在处理完已有客户端的请求前，却不会对新的客户端作出响应。

并行服务器：可以单独处理没一个连接，且不会产生干扰。并行服务器分为两种：一客户一线程和线程池。

每个新线程都会消耗系统资源：创建一个线程将占用CPU周期，而且每个线程都自己的数据结构（如，栈）也要消耗系统内存。另外，当一个线程阻塞（block）时，JVM将保存其状态，选择另外一个线程运行，并在上下文转换（context switch）时恢复阻塞线程的状态。随着线程数的增加，线程将消耗越来越多的系统资源。这将最终导致系统花费更多的时间来处理上下文转换和线程管理，更少的时间来对连接进行服务。那种情况下，加入一个额外的线程实际上可能增加客户端总服务时间。

我们可以通过限制总线程数并重复使用线程来避免这个问题。与为每个连接创建一个新的线程不同，服务器在启动时创建一个由固定数量线程组成的线程池（thread pool）。当一个新的客户端连接请求传入服务器，它将交给线程池中的一个线程处理。当该线程处理完这个客户端后，又返回线程池，并为下一次请求处理做好准备。如果连接请求到达服务器时，线程池中的所有线程都已经被占用，它们则在一个队列中等待，直到有空闲的线程可用。

 

 **** ****

**2.4写一个简单的chat程序，并能互传文件，编程语言不限。**