# 网络编程 --

1. 利用java语言开发网络相关程序

1.网络概述

## OSI七层协议：

物理层 数据链路层 网络层 传输层 会话层 表示层 应用层

IP TCP/UDP HTTP FTP POP3 SMTP telnet...

# 四层协议：

## ip地址 - ipv4：

用来标识网络中的不同主机的地址信息。 一共四位，每位0~255 。 127.0.0.1永远指向本机。主机在网络中的名称。 DNS服务器 -- 域名解析的公共服务器，可以将主机名翻译成对应的ip。

ipv6

端口号 : 每一个机器具有很多个端口号 0~65535个 端口号就像我们计算机对外打开的一扇窗户，程序可以指定对应的端口号进行监听。外界访问时除了要通过ip地址找到当前机器以外，还要通过端口号，找到对应窗口，从而找到对应程序为这次请求服务。

0~1024是系统自用的我们不能使用，另外一些常见的端口最好回避开

80

8080

25

110

DNS服务器、Hosts文件

TCP/IP 是一套协议族，包含了很多具体功能的协议，其中最重要的是IP协议、TCP协议、UDP协议。

IP协议：

## UDP协议：

将数据及源和目的地封装为数据包，不需要建立连接

每个数据报的大小限制在64k内

因为无连接，是不可靠协议

不需要建立连接，速度快

应用场景：对连接速度要求高 数据传输是否可靠不重要的情况下 使用UDP 视频聊天

## TCP协议：

建立连接，形成传输数据的通道

在连接中进行大数据量传输

三次握手，是可靠协议

必须先建立连接，效率会稍低

应用场景：对数据可靠性要求高，速度可以妥协的情况下 使用TCP 文件的传输

## 2.API介绍

InetAddress

static InetAddress getLocalHost() //用来获取本机IP信息

static InetAddress getByName(String host) //获取指定IP信息

String getHostAddress()

返回 IP 地址字符串（以文本表现形式）。

String getHostName()

获取此 IP 地址的主机名。

## 3.Socket编程 -- 套接字编程

本身是一种网络通信方式

通信的过程中两头都是Socket

利用Socket可以实现UDP 和 TCP方式的通信。

## 4.UDP编程

实验：UDP的简单通信

DatagramSocket -- udp服务

DatagramSocket()

void receive(DatagramPacket p)

从此套接字接收数据报包。

void send(DatagramPacket p)

从此套接字发送数据报包。

close();

Socket使用完后必须进行close的操作进行关闭。

DatagramPacket -- udp数据包

DatagramPacket(byte[] buf, int length, InetAddress address, int port)

InetAddress addr = dp.getAddress();//获取ip

int port = dp.getPort();//获取端口

int length = dp.getLength();//获取数据的长度

byte [] data = dp.getData();//获取具体的数据

案例：单人聊天程序

案例：群聊255.255.255.255 向这个ip发送的消息，就相当于是一个广播消息，整个网络都可以接受到

案例：多线程实现聊天

# 5.TCP编程

在TCP中分为客户端和服务器端。主动发出请求的称为客户端，被动等待请求的称为服务端。

一旦连接建立起来可以双向的进行数据的传输。

这种数据的传输是可靠的。

Socket：创建TCP的客户端

Socket连接后可以获取输出/输入的流，通过套接字以流的形式写出/读入数据。

ServerSocket：创建TCP的服务端

案例：TCP入门案例

案例：TCP双向传输数据

案例：实现大小写转换服务器