

日期: /

网络编程要素概述

1. IP

1. IP: 唯一的标识 Internet 上的计算机 (通信实体)
2. 在Java中使用InetAddress类代表IP
3. IP分类: IPv4 和 IPv6 ; 万维网 和 局域网
4. 域名: www.baidu.com www.mi.com www.sina.com www.jd.com
www.vip.com
5. 本地回路地址: 127.0.0.1 对应着: localhost
6. 如何实例化InetAddress: 两个方法: `getByName(String host)` 、 `getLocalHost()`
两个常用方法: `getHostName()` / `getHostAddress()`

2. 端口号

IP与端口号组成套接字Socket

3. TCP、UDP协议

● TCP协议:

- 使用TCP协议前, 须先建立TCP连接, 形成传输数据通道
- 传输前, 采用“三次握手”方式, 点对点通信, 是可靠的
- TCP协议进行通信的两个应用进程: 客户端、服务端。
- 在连接中可进行大数据量的传输
- 传输完毕, 需释放已建立的连接, 效率低 I

● UDP协议:

- 将数据、源、目的封装成数据包, 不需要建立连接
- 每个数据报的大小限制在64K内
- 发送不管对方是否准备好, 接收方收到也不确认, 故是不可靠的
- 可以广播发送
- 发送数据结束时无需释放资源, 开销小, 速度快

日期: /

TCP 网络编程

1. 客户端 client

```
try {  
    //1. 创建Socket对象, 指明服务器端ip和端口号  
    InetAddress inet = InetAddress.getByName("127.0.0.1");  
    socket = new Socket(inet, port: 9111);  
    //2. 获取一个输出流, 用于输出数据  
    os = socket.getOutputStream();  
    os.write("你好, 我是客户端".getBytes());  
} catch (IOException e) {  
    e.printStackTrace();  
}  
finally {
```

2. 服务端 server

```
try {  
    //1. 创建服务器端的ServerSocket, 指明自己的端口号  
    ss = new ServerSocket(port: 9111);  
    //2. 调用accept () 方法表示接收来自客户端的Socket  
    socket = ss.accept();  
    //3. 获取输入流  
    is = socket.getInputStream();  
  
    //4. 读取输入流中的数据  
    //使用BAOS防止乱码  
    byte[] cbuf = new byte[5];  
    int len;  
    baos = new ByteArrayOutputStream();  
    while ((len = is.read(cbuf)) != -1) {  
        baos.write(cbuf, off: 0, len);  
    }  
  
    System.out.println(baos);  
}
```

UDP 网络编程

日期: /

URL 网络编程

(1) 基本

① 统一资源定位符

② 格式

(2) 实例化及方法

● 一个URL对象生成后，其属性是不能被改变的，但可以通过它给定的方法来获取这些属性：

- > public String getProtocol() 获取该URL的协议名
- > public String getHost() 获取该URL的主机名
- > public String getPort() 获取该URL的端口号
- > public String getPath() 获取该URL的文件路径
- > public String getFile() 获取该URL的文件名
- > public String getQuery() 获取该URL的查询名

(3) 下载资源

```
URL url = new URL( spec: "http://localhost:8080/examples/beauty.jpg");
URLConnection urlConnection = (URLConnection) url.openConnection();
urlConnection.connect();

InputStream is = urlConnection.getInputStream();
FileOutputStream fos = new FileOutputStream( name: "beauty3.jpg");

byte[] buffer = new byte[1024];
int len;
while((len = is.read(buffer)) != -1){
    fos.write(buffer, off: 0, len);
}

//关闭资源
is.close();
fos.close();
urlConnection.disconnect();
```

