

DAY 5——UDP 协议通信

问题 5：如何实现 UDP 方式通信？

5.1 学习内容

掌握 UDP 编程，请学习教程的第 5 章 UDP 协议通信。

5.2 任务清单

- ① 运行并阅读程序，描绘出主要函数的步骤和程序结构。
- ② 学习和运用 UDP，改写一客户一线程的服务器程序。
- ③ 理解程序内部的通信协议。
- ④ 在小组内，抽签阐述服务器或客户端的软件结构，并录制视频。

5.3 关键技术

5.3.1 DatagramSocket 类（套接字）

创建 DatagramSocket

- 1. DatagramSocket(); //创建本机随机端口的 socket
- 2. DatagramSocket(int port) //指定本地端口
- 3. DatagramSocket(int port, InetAddress laddr) //指定本地 IP 和端口

DatagramSocket 类的收发函数：

- 1. .receive(DatagramPacket p)
- 2. .send(DatagramPacket p)

5.3.2 DatagramPacket 类（数据报）

1. `DatagramPacket(byte[] buf, int length)` //创建一个空数组，来构造一个 `DatagramPacket` 对象，创建输入数据报时用。
2. `DatagramPacket(byte[] buf, int length, InetAddress laddr, int port)` //同时指定 IP 和端口 输出专用。
3. `DatagramPacket(byte[] buf, int off, int length)` //从偏移量开始的长度，创建输入数据报时用。
4. `DatagramPacket(byte[] buf, int off, int length, InetAddress laddr, int port)` 输出专用。

`DatagramPacket` 类的方法函数：

1. `.getAddress();` //返回 `InetAddress`，即 IP
2. `.getport();` //返回端口
3. `.getSocketAddress();` //返回 `SocketAddress`，包含 IP 和端口
4. `.setData(buff);` //设置指定的 `buff` 为输出内容

5.3.3 应用示例

1、创建 socket

```
DatagramSocket socket = new DatagramSocket(3000);
```

2、创建输入数据报

```
byte [] inBuff=new byte [4096];
```

```
DatagramPacket inPacket=new DatagramPacket(inBuff, 4096);
```

3、创建输出数据报

```
byte [] outBuff=new byte [4096];
```

```
DatagramPacket outPacket=new DatagramPacket(outBuff,4096, InetAddress  
laddr,int port);
```

i、指定要输出的 buff

可以随时改变输出数据报中的数组，以发出不同的内容。

```
outPacket.setData(buff);
```

4、读/写数据报

```
socket.receive(inPacket); //内容存放在 inBuff 中，长度为 inPacket.getLength()  
socket.send(outPacket);
```

5、获取输入数据报中的数据

```
inPacket.getData(); //返回 byte[]
```

6、byte[]转字符串

显示 inBuff 的内容

```
System.out.println(new String (inbuff,0,inPacket.getLength())); //从 0 位置开  
始
```

7、字符串转 byte[]

```
byte [] buff=String.getBytes(); // 使用时 String 换成具体对象
```

5.4 问题讨论

- ① 如果是更加复杂的内部协议，且动作顺序不确定，如何解决？
- ② 如何实现私聊？你会采用什么方法实现？