<1>系统概述

运行环境：操作系统为Windows 编程语言python

编译：对于python程序，无需编译，直接使用python解释器运行

使用方法：使用jupyter创建python文件再运行

实现环境：需要安装python环境

程序文件列表： tcp\_server.py（TCP服务器端程序）

tcp\_client.py（TCP客户端程序）

udp\_server.py（UDP服务器端程序）

udp\_client.py（UDP客户端程序）

<2>主要数据结构

TCP数据结构：socket对象 用于建立和管理TCP连接

连接套接字 服务器端用于接收客户端连接的套接字

数据缓冲区 用于接收和发送数据的缓冲区

UDP数据结构：socket对象 用于发送和接受UDP数据包

数据缓冲区 用于储存接收的数据包

<3>主要算法描述

TCP算法描述：

服务器端：

* 1. 创建一个TCP socket。
  2. 绑定到指定的IP地址和端口。
  3. 监听传入的连接请求。
  4. 接受客户端连接。
  5. 接收客户端发送的数据。
  6. 处理数据并发送响应。
  7. 关闭连接。

客户端：

* 1. 创建一个TCP socket。
  2. 连接到服务器的IP地址和端口。
  3. 发送数据到服务器。
  4. 接收服务器的响应。
  5. 关闭连接。

UDP算法描述：

服务器端：

* 1. 创建一个UDP socket。
  2. 绑定到指定的IP地址和端口。
  3. 接收数据包。
  4. 处理数据并发送响应。
  5. 关闭socket。

客户端：

* 1. 创建一个UDP socket。
  2. 发送数据包到服务器的IP地址和端口。
  3. 接收服务器的响应。
  4. 关闭socket。

<4>用户使用手册

启动TCP服务器端程序：

python tcp\_server.py

服务器将会在指定的端口上监听连接请求。

启动TCP客户端程序：

python tcp\_client.py

客户端将会连接到服务器，并发送请求数据。

启动UDP服务器端程序：

python udp\_server.py

服务器将会在指定的端口上接收数据包。

启动UDP客户端程序：

python udp\_client.py

客户端将会发送数据包到服务器，并接收响应。

<5>程序源代码

Tcp\_client



Tcp\_server



Udp\_server



Udp\_client

