An-Najah National University

Faculty of Engineering

Computer Engineering Department

Computer networks 1

First Semester 2023/2024

Network Programming HW 1

Dr. Saed Tarapiah

Student name:

Wala’ Essam Ashqar.

ID:

12027854.

- ## Introduction:

Implementing a client server simple program, Using TCP and UDP.

- ## Discussion:

Using TCP:

1. Client Code:

A. Creating the socket in the client side.
B. Then connecting with the server by server's IP (192.168.37.1) and port number (6789).
C. Next, the client sends requests.
D. Finally, it receives the reply from the server and prints it.
E. It can ask for more requests or exit.

```java
import java.io.*;
import java.net.*;
    public class TCP_Client {
    public static void main(String argv[]) throws Exception
    {        String flag;
    do{
    String sentence;
    String output;
    System.out.println("PLease enter Vehicle plate-ID:");

    BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));

    Socket clientSocket = new Socket("192.168.37.1", 6789);

    DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());

    BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

    sentence = inFromUser.readLine();

    outToServer.writeBytes(sentence + '\n');
    System.out.println( "The sent sentence:"+sentence+'\n');

    output = inFromServer.readLine();

    System.out.println("FROM SERVER: \n");
    System.out.println("VehicleplateID   Make   Model   Year   Colour   OwnerName   OwnerID");
    System.out.println( output);

    clientSocket.close();

    System.out.println("\nPlease if you are done enter exit, if not press enter.");
                flag=inFromUser.readLine();
                clientSocket.close();

                } while(!flag.equalsIgnoreCase("exit"));}}
```

2. Server Code:

A. At first, creating the socket in the Server side with port number (6789).

B. Then create the welcome socket and process.

C. Now it is ready for any requests.

D. When a request comes, it reads it and replies.

E. The data that the server sent is stored in a data file.

```java
import java.io.*;
    public class TCP_Server{
      static int flag=0;
    public static void main(String argv[]) throws FileNotFoundException, IOException
    {   File dataFile = new File("data.txt");
    try (Scanner input = new Scanner(dataFile)) {
    HashMap<Integer, String> Vehicles = new HashMap<>();
    int lineNum = 0;
    while (input.hasNextLine()) {
        String line = input.nextLine();
        Vehicles.put(lineNum, line);
        lineNum++;}
         String clientSentence;
         String capSentence="";
         System.out.println("It works successfully!"+'\n');
         for (Map.Entry<Integer, String> entry : Vehicles.entrySet()) {
                System.out.println("Line " + entry.getKey() + " " + entry.getValue());}
        try (ServerSocket welcomeSocket = new ServerSocket(6789)) {
            while(true){
             Socket connectionSocket = welcomeSocket.accept();
             BufferedReader inFromClient = new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));

             DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());

             clientSentence = inFromClient.readLine().toUpperCase();
             System.out.println("The clientSentence:"+clientSentence+'\n');

             for (Map.Entry<Integer, String> entry : Vehicles.entrySet()) {
                 if (entry.getValue().contains(clientSentence))
                capSentence= entry.getValue();

                 else capSentence="Vehicle is not found"+'\n';
             }
             System.out.println(" The capSentence:"+capSentence+'\n');
             outToClient.writeBytes(capSentence+ '\n'); }}}}}
```

3. The text file:

255edd25-cca0-4294-909c-b 255edd25-cca0-4294-909c-b data.txt

File Edit View

VehicleplateID	Make	Model	Year	Colour	OwnerName	OwnerID
B001	Seat	Ibiza	2009	Orange	SAed	9087654
B002	Hyndai	Kona	2019	White	Sami	8750888
A001	VW	Polo	2005	black	Noora	3467892
C002	Audi	A6	2020	silver	Hiba	0956784
C003	BMW	X7	2022	brown	Noah	9076045

4. Running the TCP:

A. Server side:

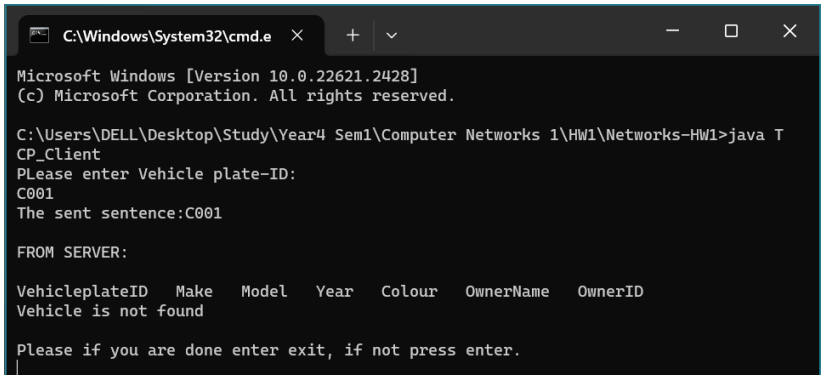
Here I printed the data from the file to be sure that he read them correctly.

```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL\Desktop\Study\Year4 Sem1\Computer Networks 1\HW1\Networks-HW1>java TCP_Server
It works successfully!

Line 0 VehicleplateID   Make   Model   Year   Colour   OwnerName   OwnerID
Line 1 B001             Seat   Ibiza   2009   Orange   SAed        9087654
Line 2 B002             Hyndai Kona    2019   White    Sami        8750888
Line 3 A001             VW     Polo    2005   black    Noora       3467892
Line 4 C002             Audi   A6      2020   silver   Hiba        0956784
Line 5 C003             BMW    X7      2022   brown    Noah        9076045
```

B. Client side:

```
C:\Windows\System32\cmd.e

Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL\Desktop\Study\Year4 Sem1\Computer Networks 1\HW1\Networks-HW1>java TCP_Client
PLease enter Vehicle plate-ID:
B001
The sent sentence:B001

FROM SERVER:

VehicleplateID   Make   Model   Year   Colour   OwnerName   OwnerID
B001             Seat   Ibiza   2009   Orange   SAed        9087654

Please if you are done enter exit, if not press enter.

PLease enter Vehicle plate-ID:
C002
The sent sentence:C002

FROM SERVER:

VehicleplateID   Make   Model   Year   Colour   OwnerName   OwnerID
C002             Audi   A6      2020   silver   Hiba        0956784

Please if you are done enter exit, if not press enter.
a001
PLease enter Vehicle plate-ID:
a001
The sent sentence:a001

FROM SERVER:

VehicleplateID   Make   Model   Year   Colour   OwnerName   OwnerID
A001             VW     Polo    2005   black    Noora       3467892

Please if you are done enter exit, if not press enter.
```

The code can find the ID if it lower or upper case, however if I enter wrong ID he will print “not found” .



Using UDP:

1. **Client Code:**

A. Creating the socket in the client side.

B. Then connecting with the server by server’s IP (192.168.37.1).

C. Next, the client sends requests using port number (9876)

D. . Finally, it receives the reply from the server and prints it.

E. It can ask for more requests or exit.

```java
import java.io.*;
import java.net.*;
class UDP_Client {
    public static void main(String args[]) throws Exception
        {
            String flag;
        do{
        System.out.println("PLease enter Vehicle plate-ID:");

        BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));

        DatagramSocket clientSocket = new DatagramSocket();

        InetAddress IPAddress = InetAddress.getByName("192.168.37.1");

        byte[] sendData = new byte[1024];
        byte[] receiveData = new byte[1024];

        String sentence = inFromUser.readLine();
        sendData = sentence.getBytes();
         System.out.println("The sentence:"+sentence+'\n');
        DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, 9876);

        clientSocket.send(sendPacket);
        DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
        clientSocket.receive(receivePacket);

        String data =new String(receivePacket.getData());
        System.out.println("FROM SERVER: \n");
        System.out.println("VehicleplateID   Make   Model   Year   Colour   OwnerName   OwnerID");
        System.out.println(data.trim());
        System.out.println("\nPlease if you are done enter exit, if not press enter.");
        flag=inFromUser.readLine();
        clientSocket.close();
        } while(!flag.equalsIgnoreCase("exit"));}}
```

2. Server code:

A. Creating the socket in the Server side with port number (9876).

B. It is ready for any requests.

C. When a request comes, it reads it and replies specifying client address, port number.

D. The data that the server sent is stored in a file.

```
public class UDP_Server {
    public static void main(String args[]) throws Exception
    {   File dataFile = new File("data.txt");
     try (Scanner input = new Scanner(dataFile)) {
     HashMap<Integer, String> Vehicles = new HashMap<>();
     int lineNum = 0;
     while (input.hasNextLine()) {
         String line = input.nextLine();
         Vehicles.put(lineNum, line);
         lineNum++;
     }
      System.out.println("It works successfully!"+'\n');
         for (Entry<Integer, String> entry : Vehicles.entrySet()) {
             System.out.println("line: " + entry.getKey() +  " "+entry.getValue());
         }
         try (DatagramSocket serverSocket = new DatagramSocket(9876)) {
             byte[] receiveData = new byte[1024];
             byte[] sendData = new byte[1024];
             while(true){
 DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
 serverSocket.receive(receivePacket);

 String sentence = new String(receivePacket.getData()).toUpperCase();
 InetAddress IPAddress = receivePacket.getAddress();
 int port = receivePacket.getPort();
  System.out.println("The sentence:"+sentence+'\n');
 String capSentence="";
  for (Map.Entry<Integer, String> entry : Vehicles.entrySet())
      if (entry.getValue().contains(sentence.trim()))
     capSentence= entry.getValue();
     //else  capSentence="Vehicle is not found"+'\n';
  System.out.println("The capSentence:"+capSentence+'\n');
       sendData = capSentence.getBytes();

       DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress,port);
       serverSocket.send(sendPacket); }}}}}
```

3. The text file:

```
255edd25-cca0-4294-909c-b 255edd25-cca0-4294-909c-b data.txt
File Edit View

VehicleplateID   Make   Model   Year   Colour   OwnerName   OwnerID
B001             Seat   Ibiza   2009   Orange   SAed        9087654
B002             Hyndai Kona    2019   White    Sami        8750888
A001             VW     Polo    2005   black    Noora       3467892
C002             Audi   A6      2020   silver   Hiba        0956784
C003             BMW    X7      2022   brown    Noah        9076045
```

4. Running the TCP:

C. Server side:

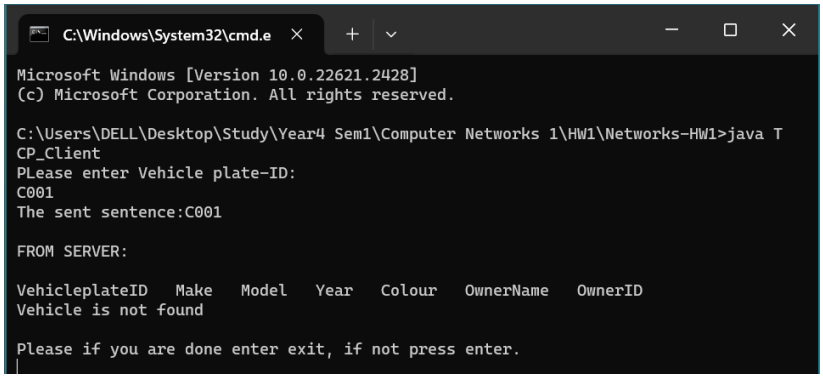
Here I printed the data from the file to be sure that he read them correctly.

```
C:\Windows\System32\cmd.e

Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL\Desktop\Study\Year4 Sem1\Computer Networks 1\HW1\Networks-HW1>java UDP_Server
It works successfully!

line: 0 VehicleplateID   Make   Model   Year   Colour   OwnerName   OwnerID
line: 1 B001             Seat   Ibiza   2009   Orange   SAed        9087654
line: 2 B002             Hyndai Kona    2019   White    Sami        8750888
line: 3 A001             VW     Polo    2005   black    Noora       3467892
line: 4 C002             Audi   A6      2020   silver   Hiba        0956784
line: 5 C003             BMW    X7      2022   brown    Noah        9076045
```

D. Client side:

```
C:\Windows\System32\cmd.e

Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL\Desktop\Study\Year4 Sem1\Computer Networks 1\HW1\Networks-HW1>java UDP_Client
PLease enter Vehicle plate-ID:
A001
The sentence:A001

FROM SERVER:

VehicleplateID   Make   Model   Year   Colour   OwnerName   OwnerID
A001             VW     Polo    2005   black    Noora       3467892

Please if you are done enter exit, if not press enter.

PLease enter Vehicle plate-ID:
b001
The sentence:b001

FROM SERVER:

VehicleplateID   Make   Model   Year   Colour   OwnerName   OwnerID
B001             Seat   Ibiza   2009   Orange   SAed        9087654

Please if you are done enter exit, if not press enter.

PLease enter Vehicle plate-ID:
c002
The sentence:c002

FROM SERVER:

VehicleplateID   Make   Model   Year   Colour   OwnerName   OwnerID
C002             Audi   A6      2020   silver   Hiba        0956784

Please if you are done enter exit, if not press enter.
```

The code can find the ID if it lower or upper case, however if I enter wrong ID he will print “not found” .



IP configuration for the network card:

IP configuration

```
C:\Users\DELL>ipconfig

Windows IP Configuration


Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::f498:79e1:14ae:c0d3%4
   IPv4 Address. . . . . . . . . . . : 192.168.37.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::3a93:8b5d:3391:a982%11
   IPv4 Address. . . . . . . . . . . : 192.168.159.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : domain.name
   Link-local IPv6 Address . . . . . : fe80::718c:ebab:e85b:3b1a%9
   IPv4 Address. . . . . . . . . . . : 192.168.1.18
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::20e:f4ff:fec5:66f2%9
                                       192.168.1.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

The name of the device:

```
The name of the device

Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL> ping -a 192.168.37.1

Pinging Wala-Essam [192.168.37.1] with 32 bytes of data:
Reply from 192.168.37.1: bytes=32 time<1ms TTL=128
Reply from 192.168.37.1: bytes=32 time<1ms TTL=128
Reply from 192.168.37.1: bytes=32 time<1ms TTL=128
Reply from 192.168.37.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.37.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

- **Conclusion:**

A. Noticing that both codes TCP and UDP give the same results.
B. but the difference is that in TCP we build a connection between the client and the server before any requests.
C. The UDP we started requesting in the beginning.

Thank you.