

Discipline: Network programming

Laboratory Work nr.5

Topic: TCP.

Done by student: _____(Untilov Andrei, gr.FAF-151)

Verified by: _____(Gavrișco Alexandru)

Chisinau 2018

Contents

1	Topic	3
2	Tasks	3
3	Theory	4
4	Task Implementation	5
5	Conclusion	7

1 Topic

TCP

2 Tasks

Develop a client-server application with the following specs:

1. Client should connect to server and allow to send commands:
 - (a) Commands have to be inserted from keyboard;
 - (b) The response have to be shown to user;
2. Server accepts conection from client through a port;
3. Server gets commands from client;
4. Server sends response to client.

Message's format:

1. Commands must start with "/" ;
2. If command accepts parameters, then parameters must be separated by " " (space).
3. If server recieves an invalid command, it must respond an informative message.
4. If server gets an invalid command , but there is a similar command , it has to return an informative message.

3 Theory

Transmission Control Protocol/Internet Protocol is a suite of communication protocols used to interconnect network devices on the internet. TCP/IP can also be used as a communications protocol in a private network (an intranet or an extranet).

TCP/IP specifies how data is exchanged over the internet by providing end-to-end communications that identify how it should be broken into packets, addressed, transmitted, routed and received at the destination. TCP/IP requires little central management, and it is designed to make networks reliable, with the ability to recover automatically from the failure of any device on the network.

TCP defines how applications can create channels of communication across a network. It also manages how a message is assembled into smaller packets before they are then transmitted over the internet and reassembled in the right order at the destination address.

IP defines how to address and route each packet to make sure it reaches the right destination. Each gateway computer on the network checks this IP address to determine where to forward the message.

4 Task Implementation

TCP/IP uses the client/server model of communication in which a user or machine (a client) is provided a service (like sending a webpage) by another computer (a server) in the network. That's why are needed two classes `Server.class` and `Client.class`. The connection between client and server is released by socket, which specifies the connection port. Both, the client and server has an object of type `Socket` with the same port, where client's socket connects to localhost (Figure 1.), while server's socket connects to `ServerSocket`.(Figure 2.)

```
public class Client {  
  
    public static void main(String[] args) throws UnknownHostException,  
  
        InetAddress host = InetAddress.getLocalHost();  
        Socket socket = null;  
        ObjectOutputStream oos = null;  
        ObjectInputStream ois = null;  
        socket = new Socket(host.getHostName(), 9876);  
  
        Scanner sc=new Scanner(System.in);  
        ois=new ObjectInputStream(socket.getInputStream());  
        System.out.println(ois.readObject().toString());  
        while(true){  
            System.out.print("Input command : ");  
            String command=sc.nextLine();  
            oos = new ObjectOutputStream(socket.getOutputStream());  
            oos.writeObject(command);  
            if(command.equals("/exit")) break;  
            ois = new ObjectInputStream(socket.getInputStream());  
            String response = (String) ois.readObject();  
            System.out.println("Bot: " + response);  
        }  
        ois.close();  
        oos.close();  
        socket.close();  
    }  
}
```

Figure 1. - Client.java

```

public class Server{

    private static ServerSocket server;
    private static int port = 9876;
    private static String recievedCommand;
    private static String recievedArgument;

    static String[] commandsList={"/hello","/day","/time","/random","/exit"};
    static String[] daysOfWeek={ "Monday","Tuesday", "Wednesday", "Thursday", "Friday","S

    public static void main(String args[]) throws IOException, ClassNotFoundException{
        server = new ServerSocket(port);
        ObjectInputStream ois=null;
        ObjectOutputStream oos=null;
        Socket socket=null;
        socket = server.accept();
        oos = new ObjectOutputStream(socket.getOutputStream());
        oos.writeObject("Input '/help' command to get the list of supported commands.");
    }
}

```

Figure 2. - Server.java

The communication between server and client is performed using ObjectInputStream.java (reading data) and ObjectOutputStream.java (writing data) classes. When client writes data to server using ObjectOutputStream, server reads data using ObjectInputStream and vice-versa.

***The link to repository : https://github.com/untilovandrei/PR_Lab_4.git**

5 Conclusion

During this laboratory work has been obtained basic experience of working with TCP/IP protocol. It gives us the opportunity to build a flexible architecture . Also the property of connections to remain intact as long as the source and destination machines are functioning.