

TP3

commande d'activation TCP --> nc -l -p 2000

commande dans l'autre console --> nc 127.0.1 2000

-l --> permet d'activer le mode lecture

-p --> permet de fixer le numéro de port (ici 2000)

On peut analyser la trame avec WireShark

oui il l'est car il faut d'abord mettre l'une des console en réception

Avec ce protocole de communication, la connexion par client direct n'est pas possible

Exercice 2 : Envoi de segment TCP avec un programme simple

cf. code Client TCP dans le dossier TD3

Exercice 2

Réalisez un programme qui :

- ouvre une socket TCP vers une adresse IP et un port particulier
- envoie un message simple à l'intérieur de la socket
- ferme la connexion dès qu'il a envoyé les données, sans lire de réponse en retour.

Vérifiez le bon fonctionnement de votre programme en utilisant netcat pour afficher les données envoyées par votre programme (netcat fera office de serveur TCP).

Exercice 3 -> attend une demande de connexion TCP

```

1  package TD3;
2
3  import java.io.IOException;
4  import java.io.InputStream;
5  import java.io.OutputStream;
6  import java.net.InetAddress;
7  import java.net.InetSocketAddress;
8  import java.net.Socket;
9
10 /**
11  * Client basique TCP
12  *
13  */
14 public class ClientTCP
15 {
16
17     public static void main(String[] args) throws Exception
18     {
19         ClientTCP clientTCP = new ClientTCP();
20         clientTCP.execute();
21     }
22
23     /**
24     * Le client cree une socket, envoie un message au serveur
25     * et attend la reponse
26     *
27     */
28     private void execute() throws IOException
29     {
30         //
31         System.out.println("Demarrage du client ...");
32
33         //Creation de la socket
34         Socket socket = new Socket();
35
36         // Connexion au serveur
37         InetSocketAddress adrDest = new InetSocketAddress("127.0.0.1", 7000);
38         socket.connect(adrDest);
39
40         // Envoi de la requete
41         byte[] bufE = new String("question du client ?\n").getBytes();
42         OutputStream os = socket.getOutputStream();
43         os.write(bufE);
44         System.out.println("Message envoye");
45         System.out.println("IP:"+socket.getInetAddress());
46         System.out.println("Port:"+socket.getPort());
47
48         // Attente de la reponse
49         byte[] bufR = new byte[2048];
50         InputStream is = socket.getInputStream();
51         String reponse = "";
52
53         while (!reponse.startsWith("close")) {
54
55             int lenBufR = is.read(bufR);
56             if (lenBufR!=-1)
57             {
58                 reponse = new String(bufR, 0 , lenBufR );
59                 System.out.println("Reponse recue = "+reponse);
60             }
61         }
62
63         // Fermeture de la socket
64         socket.close();
65         System.out.println("Arret du client .");
66     }
67 }
68
69

```