ESISAR

NE449 - Programmation répartie

TDM numéro 1

SOLUTIONS

Table des matières

1 Solution 1

Pour utiliser l'outil netcat dans un terminal vous devez taper la commande “netcat” ou son raccourci “nc”.

Pour connaître la liste des options disponibles pour cet outil, vous pouvez consulter l'aide en tapant dans un terminal

```
nc -help
```

Pour échanger des messages UDP

Ouvrir un terminal et taper

Terminal1

```
nc -l -u -p 2000
```

Ce terminal 1 va écouter sur le port 2000 en utilisant UDP (l'option -u). L'option -l permet de savoir que l'on écoute. Maintenant ouvrez un terminal 2 et taper :

Terminal2

```
nc -u 127.0.0.1 2000
```

Nous utilisons l'adresse de loopback (127,0,0,1) car nous utilisons deux terminaux sur la même machine. Maintenant vous pouvez envoyez un message, en le saisissant dans le terminal 2 et en faisant entrée. Il devrait alors apparaître dans le terminal 1.

Netcat utilise l'appui sur la touche entrée pour savoir qu'il doit envoyer un paquet UDP. Avant l'appui sur entrée, il ne fait rien.

Wireshark

Utilisez wireshark pour observer les paquets. Vous noterez que vous devez être root pour utiliser Wireshark.

L'ordre de lancement.

L'ordre de lancement n'a pas d'importance, car aucun paquet n'est échangé avant l'envoi sur entrée.

2 Solution 2

Pour échanger des messages UDP avec une autre machine

Sur la machine 1, ouvrir un terminal et taper

Machine1

```
nc -l -u -p 2000
```

Sur la machine 2, ouvrir un terminal et taper

Machine2

```
nc -u xxxx 2000
```

Vous devez ici remplacer xxxx par l'adresse IP de la machine 1. Pour connaître l'adresse IP de la machine 1, en tant que root, il faut saisir la commande ifconfig.

3 Solution 3

Voici le programme à réaliser :

```
package tdm1.exo3;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetSocketAddress;

/**
 * Client basique UDP
 */
public class ClientUDP
{

	public static void main(String[] args) throws Exception
	{
		ClientUDP clientUDP = new ClientUDP();
		clientUDP.execute();

	}


	/**
	 * Le client cree une socket, envoie un message UDP au serveur
	 */
	private void execute() throws IOException
	{
		//
		System.out.println("Demarrage du client ...");

		//Creation de la socket
		DatagramSocket socket = new DatagramSocket();

		// Creation et envoi du message à l'adresse 127.0.0.1 et le port 2000
		InetSocketAddress adrDest = new InetSocketAddress("127.0.0.1", 2000);
		byte[] bufE = new String("hello").getBytes();
		DatagramPacket dpE = new DatagramPacket(bufE, bufE.length, adrDest);
		socket.send(dpE);
		System.out.println("Message envoyé.");


		// Fermeture de la socket
		socket.close();
		System.out.println("Arret du client .");
	}

}
```

Dans ce programme, l'adresse IP de destination et le port sont en dur (127.0.0.1 et 2000).

Pour tester : vous lancez dans un terminal la commande

```
nc -l -u -p 2000
```

Vous lancez ensuite votre programme Java.

Normalement, le mot « hello » doit s'afficher dans le terminal.

Vous noterez que la principale difficulté du programme vient de la transformation d'un String en un buffer. Vous étudierez en particulier ces deux lignes :

```
byte[] bufE = new String("hello").getBytes();
DatagramPacket dpE = new DatagramPacket(bufE, bufE.length, adrDest);
```

4 Solution 4

Voici le programme à réaliser :

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetSocketAddress;

/**
 * Serveur basique UDP
 */
public class ServeurUDP
{

	public static void main(String[] args) throws Exception
	{
		ServeurUDP serveurUDP = new ServeurUDP();
		serveurUDP.execute();

	}



	private void execute() throws IOException
	{
		//
		System.out.println("Demarrage du serveur ...");

		// Le serveur se declare aupres de la couche transport
		// sur le port 3000
		DatagramSocket socket = new DatagramSocket(null);
		socket.bind(new InetSocketAddress(3000));

		// Attente du premier message
		byte[] bufR = new byte[2048];
		DatagramPacket dpR = new DatagramPacket(bufR, bufR.length);
		socket.receive(dpR);
		String message = new String(bufR, dpR.getOffset(), dpR.getLength());
		System.out.println("Message recu = "+message);
```

```
            System.out.println("IP = "+dpR.getAddress().getHostAddress());
            System.out.println("Port = "+dpR.getPort());


            // Fermeture de la socket
            socket.close();
            System.out.println("Arret du serveur .");
      }

}
```

Dans ce programme, l'adresse IP d'écoute est « toutes » et le port d'écoute est en dur (3000).

Pour tester :

- vous lancez ensuite votre programme Java.
- vous lancez dans un terminal la commande

```
nc -u -p 3000
```

vous taper le mot hello dans le terminal, il doit apparaître dans la console Java

5 Solution 5

Il suffit de bien accorder le numéro de port et les adresses IP