

Exercice 1. Client/Serveur en UDP

1.1 Ecrire un programme client-serveur en utilisant cette fois les sockets en UDP. Le client envoie sa requête (peut importe le contenu), et le serveur renvoie la date actuelle au client, qui l'affiche avec l'adresse du serveur

1.2 Faire évoluer le programme. Le serveur envoie une réponse différente en fonction de la requête du client :

- date : la date actuelle est envoyée
- os : le système d'exploitation
- user : le nom d'utilisateur

La requête du client est donnée en argument du programme. Si la requête est différente des trois précédentes, un message d'erreur est envoyé par le serveur.

Exercice 2. Broadcast

2.1 Faire un serveur qui envoie en broadcast la date du jour, toutes les secondes. Faire un client qui récupère et affiche cette date.

Rappels :

Constructeur de paquet à envoyer :

```
new DatagramPacket  
    (buffer,buffer.length,  
     InetAddress.getByName("127.0.0.1"),port)
```

`new DatagramPacket(buffer, buffer.length)` : Constructeur de paquet en réception

`new DatagramSocket(port)` : Constructeur de Socket UDP

`main(String args[])` : `args[0]` premier argument, `args[1]` deuxième argument ...

`dP.getData()` : récupère à partir d'un `datagramPacket` (`dP`) la donnée

Réseau (TD n°2)

`dP.getAddress()` : récupère l'adresse de l'expéditeur d'un datagramPacket (dP)

`Date date = new Date()` : retourne la date actuelle (la méthode `toString()` permet de la convertir en chaîne de caractères)

`System.getProperty("os.name")` : retourne le système d'exploitation (chaîne de caractères)

`System.getProperty("user.name")` : retourne le nom d'utilisateur (chaîne de caractères)

`"chaîne".getBytes()` : convertit une chaîne de caractère en tableau d'octets

`new String(byte[] bytes)` : créer une chaîne de caractères à partir d'un tableau d'octets