Sockets

F. Mallet

miage.m1@gmail.com

http://deptinfo.unice.fr/~fmallet/

Les Sockets en Java : `java.net`

- Dans les systèmes embarqués
  - Les cartes Ethernet remplacent de plus en plus les liaisons série ou parallèle
- La ***socket*** (prise) est l'extrémité de communication
  - Une *socket* de chaque côté (Client-Serveur)
- **Protocoles de transport**
  - Fiable (assure l'arrivée et l'ordre d'arrivée) par octets (e.g., TCP)
  - Fiable par paquets : une opération par paquet
  - Non-fiable par paquets (e.g., UDP)

Université Nice Sophia Antipolis

Adresse IP et port

- Sur un réseau, chaque terminal a **une adresse réseau unique** dans un espace de noms
  - e.g. IPv4 : 32 bits, 127.0.0.1 = localhost
- Une *socket* est associée à l'adresse de son **hôte**
  - Elle permet d'écouter ou d'écrire sur le réseau
  - Plusieurs *sockets* sur le même hôte : numéro de port
- **Numéro de port** (entier de 0 à 65535)
  - Certains ports sont réservés pour des protocoles applicatifs particuliers (HTTP:80, SSH:22, FTP:21)

Université Nice Sophia Antipolis

La classe `ServerSocket`

- Représente la prise côté serveur
  - Accepte des connexions de clients, répond à leurs requêtes
- **`ServerSocket(int port)`** - constructeur
  - Prise qui écoute sur le port donné en paramètre
- **`Socket accept()`**
  - Se met en attente d'un connexion d'un client
  - **`setSoTimeout(int)`** permet de choisir le temps maximum d'attente, 0 = infini
  - La *socket* résultat permet de communiquer avec le client
- **close()** – Arrête le serveur

Université Nice Sophia Antipolis

La classe `Socket`

- Connexion par octets, fiable (TCP)
- Constructeur

```
Socket(InetAddress address, int port)
```

  - Tente une connexion au serveur *address:port*
- Du côté serveur comme du côté client on manipule un objet de type **`Socket`**
- Méthodes
  - flot d'octets d'entrée : lire sur la *socket*

```
InputStream getInputStream()
```

  - Renvoie un flot d'octets de sortie : écrire sur la *socket*

```
OutputStream getOutputStream()
```

  - ferme la *socket*

```
close()
```