

## TP 3 – Client et serveur TCP en Java

### Programmation réseaux, concurrente et distribuée

L'objectif de ce TP est de programmer un serveur FTP en JAVA à l'aide de sockets en mode TCP.

Au lancement du serveur, on spécifie en ligne de commande :

- Le port sur lequel écoute le serveur
- SERVEUR\_DIR (optionnel) : la racine du système de fichier que gère le serveur.

Initialement, SERVEUR\_DIR = *le répertoire courant du serveur/Downloads*

Ce serveur répond aux requêtes suivantes :

- GET\_FILE : Permet de récupérer, du répertoire SERVEUR\_DIR, un fichier identifié par son nom.
- PUT\_FILE : Permet d'écrire dans le répertoire SERVEUR\_DIR un fichier identifié par son nom du côté du client. Cette requête contient le nom du fichier à *uploader*. Le serveur retourne au client un message qui acquitte que l'upload du fichier sur le serveur a bien été effectué ou non.
- LS\_DIR : Permet de lister le contenu du répertoire SERVEUR\_DIR sur le serveur.

Le client prend en paramètre de lancement :

- Le nom de la machine du serveur
- Le port du serveur FTP
- CLIENT\_DIR = la racine du système de fichier du client

Ensuite, le même client FTP pourra servir à lancer plusieurs commandes FTP successives, sous forme texte, sur la même connexion TCP. Les commandes que l'on peut lancer en ligne de commande dans ce client FTP sont les suivantes :

- LS : permet de lister le contenu du répertoire SERVEUR\_DIR sur le serveur via la requête LS\_DIR et affiche le résultat.

- GET <nom\_fichier> : récupère le fichier à distance sur le serveur via la requête GET\_FILE, stocke le fichier localement dans le répertoire CLIENT\_DIR, et affiche le résultat.
- PUT <nom\_fichier> : envoie le fichier, présent localement dans CLIENT\_DIR, au serveur via la requête PUT\_FILE, stocke le fichier sur le serveur et affiche le résultat.

### Utilisation

On lance le serveur : *java ServerFTP 5600*

On lance un client : *java ClientFTP localhost 5600 Downloads1*. Ensuite, on pourra lancer plusieurs commandes FTP dans la même session FTP. Par exemple :

- GET toto.txt :  
Le client récupère le fichier toto.txt du serveur et le stocke localement dans le répertoire Downloads1.
- PUT popo.txt :  
Le client envoie le fichier popo.txt, du répertoire Downloads1, au serveur qui le stocke dans SERVEUR\_DIR.