

# TP de Réseaux – RICM 4

## Programmation d'une application de transfert de fichiers en utilisant les sockets

Pascal Sicard

24 novembre 2017

## 1 Introduction

Nous allons développer une application permettant le transfert de fichiers entre deux machines. Nous utiliserons le langage C et la librairie des sockets. Une documentation synthétique des sockets et des fonctions nécessaires au développement de ce programme vous est fournie.

## 2 Cahier des charges

### 2.1 Utilisations des programmes

L'utilisateur lance le client en fournissant l'adresse (ou le nom associé) et le port du serveur avec lequel il veut faire le transfert de fichier. Exemple : *transfertcl mandelbrot 9999*

Le serveur doit bien sûr être à l'écoute des demandes des clients, nous développerons un serveur multi-client permettant de dialoguer avec plusieurs clients en parallèle. Le serveur sera lancé en fournissant un numéro de port : Exemple : *transfertse 9999*

### 2.2 Fonctionnalités

Une fois le client lancé, l'utilisateur peut entrer les commandes suivantes :

- *ls* : il obtient la liste des fichiers sous le répertoire courant.
- *put nomfichier* : le fichier de nom *nomfichier* existant sur la machine du client est transféré et sauvegardé sous le même nom sur la machine serveur. Si le fichier n'est pas présent un message d'erreur est affiché.
- *get nomfichier* : le fichier de nom *nomfichier* existant sur la machine du serveur est transféré et sauvegardé sous le même nom sur la machine client. Si le fichier n'est pas présent un message d'erreur est affiché.
- *quit* : le programme client se termine.
- Dans le cas où une autre chaîne de caractères est entrée par l'utilisateur un message d'erreur s'affiche.

## 2.3 Contraintes

- Pour des raisons de fiabilité, les échanges se feront à l'aide du protocole TCP.
- Le serveur doit rester à l'écoute des demandes de différents clients. Nous développerons un serveur multi-client permettant de dialoguer avec plusieurs clients en parallèle.

## 3 Conseils de réalisation

- Si on n'a pas d'idée, on peut s'inspirer du vrai *ftp* en observant les paquets échangés lors de quelques commandes.
- On s'appliquera à écrire une application qui puisse être facilement enrichie par d'autres commandes (comme le vrai *ftp*) : *cd*, *mkdir*, *pwd*...
- On ne se soucie pas des accès à un compte utilisateur particulier. Les fichiers accessibles côté serveur sont ceux sur lequel le serveur est lancé. Le client ne fournit pas de login et mot de passe.

## 4 Compte-rendu

Vous devrez fournir comme compte rendu :

- Le code source du programme client et serveur fortement commenté.
- Une documentation annexe sur le code si cela vous semble nécessaire.
- Éventuellement les restrictions de vos programmes par rapport au cahier des charges initial.
- Des jeux de tests.
- Éventuellement des commentaires sur les problèmes rencontrés.

## 5 Références Bibliographiques

1. Introduction au C :  
[http://www-clips.imag.fr/commun/bernard.cassagne/Introduction\\_ANSI\\_C.html](http://www-clips.imag.fr/commun/bernard.cassagne/Introduction_ANSI_C.html)
2. TCP/IP : Architecture, protocoles, applications Douglas COMER, InterEditions, 1992
3. Internetworking with TCP/IP. Vol III : Client-Server programming and applications, BSD Socket Version - Douglas COMER and W. Richard STEVENS, Prentice Hall, 1993