

encadrés par Christian Attiogbé

Cahier d'exercices - FTP

Modélisation et programmation du comportement d'un serveur FTP

Le protocole *FTP* (*File Transfer Protocol*) est un protocole de transfert de fichiers entre machines situées sur un réseau. Une machine (dite **client ftp**) va déposer/charger des données situées sur une autre machine (dite **serveur ftp**).

On veut analyser le comportement d'un **client ftp** et d'un **serveur ftp**. Ces comportements correspondent à des programmes appelés ici processus.

Un processus serveur ftp attend des demandes de connexions (*open*) de processus clients, puis interagit avec eux pour rendre des services.

Un processus client, avant de charger/déposer des données, demande une connexion à un serveur ftp. Lorsque la connexion est effectuée, le serveur demande un mot de passe au client et vérifie la validité du mot de passe ; en cas de succès, le client est autorisé à effectuer des actions (décrites ci-après). En cas d'échec le client n'est pas autorisé à effectuer des actions, il peut redonner le mot de passe si le serveur le réclame.

Les actions effectuées par un processus client sont *ls*, *get*, *put*, *close* ; ce sont des actions attendues par le serveur après la connexion et la validation du mot de passe du client ;

- *ls* : lorsque le serveur reçoit cette demande, il renvoie une donnée au client, le client affiche cette donnée puis peut effectuer une autre action ;
- *get* : lorsque le serveur reçoit cette demande, il renvoie une donnée au client, le client affiche cette donnée puis peut effectuer une autre action ;
- *put* : lorsque le serveur reçoit cette demande, il renvoie un acquittement au client ; le client peut en fonction de l'acquiescement recommencer la même action *put* ou effectuer une autre action ;
- *close* : lorsque le serveur reçoit cette demande, il ferme la connexion en cours avec le client ; le client ne peut alors plus interagir avec le serveur. Le serveur lui, se met en attente d'une autre connexion.

En faisant abstraction du traitement des fichiers (les données sont envoyées de façon abstraite),

1. Concevoir à l'aide de Réseau de Petri et programmer en langage GO le programme qui implante le serveur, un client ;
2. généraliser à *n* clients ;
3. généraliser à *n* serveurs.