

Implémentation d'un pseudo serveur FTP

L'objectif du TP est de construire un serveur de fichiers inspiré par le protocole FTP

Travail Réalisé

La découpe du fichier est réalisée grâce à une constante représentant la taille du buffer. Cette taille est une constante commune au client et au serveur. Elle est située dans le fichier ConstanteFTP.c qui répertorie tout ce qui est commun aux deux parties.

La gestion des pannes est réalisée de manière très simple, si le fichier demandé au serveur via `get <fichier>` est présent dans le répertoire courant du client, alors on ne transfère que la différence entre le fichier présent sur le client et le serveur. Cette simplicité présente notamment un problème : Si le fichier du serveur a été modifié entre la panne et la redemande du fichier par le client, alors le client disposera alors d'un fichier composé de deux parties chronologiquement différentes. De plus, si la taille du fichier modifié sur le serveur est inférieure ou égale à la taille du fichier déjà téléchargé par le client, alors le transfert s'arrêtera.

L'équilibrage de charge n'a pas été implémenté

Le client peut faire plusieurs demandes de fichier par connexion (ainsi que n'importe quelle autre commande implantée) sans couper la connexion au serveur. Les deux parties bouclent jusqu'à que la commande `bye` soit engagée.

Concernant l'implémentation de toutes les commandes, aucune utilisation de `execvp` n'a été réalisée.

- La commande `ls` envoie au client : les permissions du fichier/dossier, le type (fichier/répertoire) et le nom de l'élément.
- La commande `pwd` affiche le chemin courant du serveur FTP
- La commande `cd` change le dossier courant sur le serveur FTP
- La commande `mkdir` permet de créer un dossier sur le serveur FTP
- La commande `rm` permet de supprimer un fichier sur le serveur FTP
- La commande `rm -r` permet de supprimer un dossier sur le serveur FTP, sauf qu'au lieu de supprimer les éléments, cette commande affiche tous les éléments qui auraient dû être supprimés ainsi que leur ordre. Cela évite de s'exposer à un bug de la fonction qui pourrait être dramatique
- La commande `put` permet de téléverser un fichier sur le serveur FTP, elle utilise le même protocole que pour le `get` excepté la gestion de panne.

Il est possible de s'authentifier afin d'accéder au serveur, pour l'instant seulement avec l'identifiant `user:admin`.

Des messages d'erreurs apparaissent en cas de problème de traitement de la requête.

Dans le cas d'un SIGINT sur le serveur, les fils sont kill. Concernant le client, il envoie la commande `bye` au serveur avant de se fermer

Tests Réalisés

Commande get : Vérification de la même taille et du même contenu que le fichier du serveur

Commande ls : Vérification que tous les éléments sont affichés

Commande pwd : Vérification que le chemin est bien le bon

Commande cd : Vérification du bon changement de répertoire avec ls et pwd

Commande mkdir : Vérification de la bonne création du répertoire avec ls

Commande rm : Vérification de la bonne suppression avec ls

Commande rm -r : Vérification de l'ordre de suppression avec un affichage

Commande put : même vérification que pour get

L'authentification échoue si le nom d'utilisateur ou le mot de passe n'est pas bon

Si le serveur reçoit un SIGINT alors, avec l'utilisation de la commande ps on remarque que les processus fils ont été arrêté.

Pour la panne, le début du fichier que l'on veut télécharger est copié dans le fichier de destination du client, puis on lance la commande. Seule la partie manquante est téléchargée.

Utilisation

Pour utiliser le serveur et son client, il suffit de compiler les exécutables avec le makefile, et d'utiliser pour le serveur l'identifiant user:admin