

TP FTP

Le but de ce TP est de programmer un client FTP simplifié en C, sous linux. Il se décompose en un programme de base (qui vous assure la moyenne s'il fonctionne correctement), et des parties optionnelles qui vous rapportent des points supplémentaires.

Programme de base (guidé)

Dans cette section, nous allons spécifier les détails d'implémentation nécessaires à la partie de base. Pour vérifier que tout fonctionne correctement, n'hésitez pas à tester votre programme, à capturer les trames et à les comparer à celles obtenues lorsque vous utilisez la commande Linux `ftp`.

Travail demandé

Le programme de base affiche une invite permettant de saisir des commandes et d'effectuer les tâches suivantes.

Connexion et authentification

- ouverture d'une connexion TCP de contrôle vers le port 21 du serveur FTP (commande `open <@IP>`)
- transmission des login et mot de passe via une saisie au clavier

Affichage du contenu du répertoire courant

- demander l'obtention du contenu du répertoire (commande `dir`)
- transfert des données : vous utiliserez le mode actif
- affichage du listing du répertoire à l'écran

Transfert d'un fichier texte

- demander l'obtention du contenu du fichier via la connexion de contrôle (commande `get <fichier>`)
- écriture du contenu du fichier dans un fichier local

Fermeture de la connexion

- fermeture de la connexion (commande `ciao`)

Arrêt de l'application

- terminer le programme (commande `exit`)

Code fourni

Lors de ce TP, une portion de code vous est fournie. Celle-ci permet d'effectuer correctement la lecture des commandes sur l'entrée standard. Pour cela, la librairie `readline` est utilisée. Installez donc le package `libreadline-dev`. La boucle principale du main fourni lit les commandes passées par l'utilisateur et les décompose dans les variables `argc_cmd` et `argv_cmd`, qui fonctionnent de la même manière que `argc` et `argv` que vous avez pu voir souvent. Vous allez donc devoir modifier la fonction `process_command`, afin d'être en mesure de traiter toutes les commandes requises. Le prototype des fonctions n'est pas fourni, vous pouvez le modifier à votre guise.

Les sections suivantes vous proposent un guide afin de réaliser la partie de base de votre programme.

Initialisation de la connexion et identification

Vous allez devoir réaliser une fonction `open_connexion`, qui va initialiser la connexion TCP avec le serveur. Cette fonction devra être appelée lorsque la commande `open @IP` est tapée par l'utilisateur. Cette fonction prendra donc en argument l'adresse IP du serveur, devra créer le socket, et le connecter à l'adresse en question.

Ensuite cette fonction demandera à l'utilisateur d'entrer son login, puis ensuite son mot de passe. Notez que cette étape doit être répétée tant que le couple login/password n'est pas correct.

Une fois que toutes ces étapes sont terminées, la fonction devra renvoyer le socket ainsi initialisé.

Envoi du port pour réception des données

Dans cette section vous allez devoir réaliser une fonction `send_data_port` qui permettra au serveur de savoir sur quel port les données des commandes devront être envoyées. Vous allez donc devoir créer un socket qui écoutera les connexions entrantes sur ce port, puis envoyer la commande adéquate au serveur. La fonction retournera le socket ainsi créé.

Affichage du répertoire courant

Dans cette partie du sujet, vous allez devoir créer une fonction permettant à votre client de recevoir le contenu du répertoire courant. Cette fonction sera appelée lorsque l'utilisateur tapera la commande `dir`.

Dans un premier temps, vous allez devoir spécifier sur quel port recevoir les données, et donc utiliser la fonction créée précédemment. Ensuite il faudra envoyer au serveur la commande correspondant à l'affichage du répertoire courant. Pour cette fonction et la suivante, pensez à appeler les différentes composantes dans le bon ordre, et à fermer les bons sockets.

Téléchargement d'un fichier

Dans cette partie du sujet, vous allez devoir créer une fonction permettant à votre client de télécharger un fichier depuis le serveur. Cette fonction sera appelée lorsque l'utilisateur tapera la commande `get filename`.

Dans un premier temps, vous allez devoir spécifier sur quel port recevoir les données, et donc utiliser la fonction créée précédemment. Ensuite il faudra envoyer au serveur la commande correspondant à la demande de téléchargement du fichier.

Parties optionnelles

Mode debug (+1 point)

- affichage ou non des réponses envoyées par serveur sur la connexion de contrôle,
- le basculement entre les deux modes sera possible via les commandes `debugon` et `debugoff`

Gestion des modes actif et passif (+3 points)

- permettre le transfert de données dans les modes actif ou passif, au choix de l'utilisateur,
- le basculement entre les deux modes sera possible via les commandes `passiveon` et `passiveoff`

Gestion des fichiers locaux (+4 points)

- transférer un fichier local sur le serveur FTP (commande `send <fichier>`),
- attention, il peut être alors nécessaire de préciser le mode de transfert (type ASCII ou BINARY)

Gestion des fichiers distants (+2 points)

- renommer un fichier distant (commande `ren <fichier1> <fichier2>`),
- supprimer un fichier distant (commande `del <fichier>`),

Gestion des répertoires distants (+4 points)

- se déplacer dans un répertoire distant (commande `cd <rep>`),
- remonter dans le répertoire parent (commande `cd ..`),
- créer un répertoire distant (commande `mkd <rep>`),
- supprimer un répertoire distant (commande `rmd <rep>`),
 - attention, il faut éventuellement gérer le fait que le répertoire distant à supprimer contient des fichiers

Rendre le TP

Il n'est pas demandé de rapport, mais vous devez commenter généreusement votre code.

Le travail (fichier(s) C et fichier makefile) doit être terminé à la fin de la séance et posté sur moodle, sous la forme d'une archive dont le nom sera composé des noms des membres du groupe. Vous y préciserez (fichier README) quelles parties optionnelles sont implémentées.

Il s'agit d'un travail en binôme, et toute ressemblance avec un programme trouvé sur le web ou rendu par autre groupe aboutira à une note de 0/20.