

Section	<b>I4</b>
Date	<b>26/11/2014</b>
Enseignant	<b>M. MALDONADO</b>
Matière	<b>PROJET MULTI THREADS</b>

## SERVEUR FTP

Il s'agit dans ce projet de programmer un serveur ftp en version multi-thread. Ce serveur aura pour tâche de répondre aux requêtes de transfert de fichiers émises par les clients.

Sont à écrire :

- le code du client : permettant de sélectionner les fichiers à envoyer/recevoir et suivre l'avancée du transfert.
- le code du serveur : permettant de prendre en charge les demandes des clients

La couche réseau (communication client/serveur) ainsi que le transfert effectif, même s'ils ne constituent pas le cœur de la partie multi-threading du projet, sont demandés.

En revanche la connexion ne se fera que sur un seul serveur, et le dialogue (protocole) pourra s'inspirer du protocole FTP sans le suivre à la lettre (voir [http://fr.wikipedia.org/wiki/File\\_Transfer\\_Protocol](http://fr.wikipedia.org/wiki/File_Transfer_Protocol))

## LE CLIENT

---

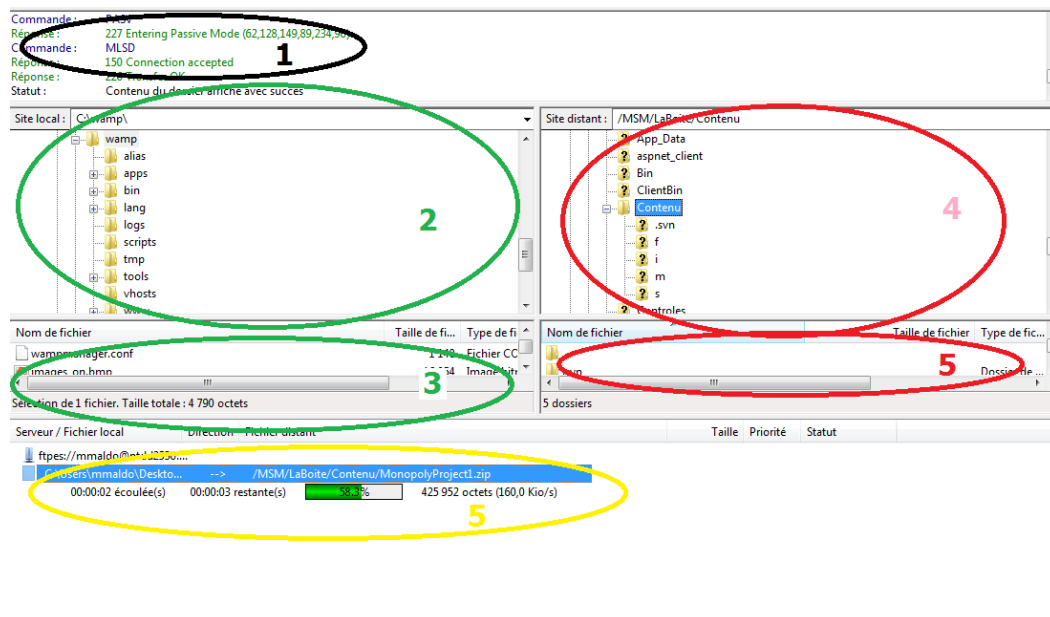
Le client après connexion au serveur, émet les demandes de transferts de fichiers vers le serveur, montant (upload) ou descendants (download).

Il doit présenter les fonctionnalités suivante:

- une page de paramétrage (adresse IP et port du serveur, identifiant de connexion)
- la possibilité de parcourir (browser) les disques locaux ou distants (du serveur) et la sélection d'un ou plusieurs fichier
- la possibilité de suivre l'avancée d'un transfert (barre de progression)
- la possibilité d'annuler un transfert

Vous pouvez vous inspirer de clients FTP classiques (cf écran) sur lesquels on distingue :

- une zone (1) affichant le dialogue FTP avec le serveur (commandes échangées)
- deux zones permettant de naviguer sur les disques locaux (dossiers 2 et fichiers 3)
- deux zones permettant de naviguer sur les disques distants (dossiers 4 et fichiers 5)
- une zone permettant de suivre l'avancée des transferts (5)



fenetre client ftp

Le client est un programme Windows Form (ou équivalent). Chaque fichier transféré sera pris en charge par un thread que créera le client

## LE SERVEUR

Le programme serveur tourne en tâche de fond sur le serveur (service). Il doit répondre aux demandes des clients. Sa première tâche est de vérifier la connexion du client (authentification).

*Rq : Vous choisirez un enregistrement des identifiants à votre convenance, mais suffisamment souple pour permettre leur consultation/modification sans recompiler ni relancer l'application serveur (base de données, fichier XML, fichier texte,...)*

Chaque client sera traité par un nouveau thread que lancera le serveur.

Le serveur doit pouvoir afficher la liste des connexions (avec leurs caractéristiques). Il doit pouvoir également clore une connexion avec un poste client.

L'interface du serveur sera laissée à la convenance du groupe d'étudiant (interface en ligne de commande, windows ou web)

## COMMUNICATION CLIENT/SERVEUR

La communication CLIENT/SERVEUR se fera par le biais de bibliothèques de composant prenant en charge la couche TCP/IP

## A RENDRE

A faire par groupe de 4 étudiants. A rendre :

- lien vers le repository des sources (github ou équivalent)
- documents d'analyse (approche UML appréciée)

Soutenance prévue

