

Younoussa TOURE

Java avancée – Simple FTP

Multi-thread the server

Afin que le programme accepte plusieurs connexions client je crée un nouveau thread dans FTPServerMain pour chacun des clients afin qu'il puisse communiquer sans encombre.

```
new Thread(new FTPServer(socketaccueil.accept())).start();
```

Pour ce faire le server implémente l'interface Runnable et plus Callable<Void>. Cette manipulation aurait très bien pu marcher avec Callable mais l'adaptation multi-thread était plus légère avec Runnable. Sachant que ces deux interfaces font la même chose excepté que la méthode call() de Callable permet de déterminer des valeurs de retour. Ce dont on a pas besoin ici.

Displays the file hierarchy on the server

Ici il était possible de créer l'arbre du côté du serveur, de le passer au client via un objet File, puis recréer l'arbre du côté du serveur. Cependant cette manipulation marche quand on travaille en local. Mais via le réseau il n'est pas cohérent d'envoyer un objet File.

Dans le case LIST de run() de FTServer : *sendSerializedObject(rootNode);*

C'est pour cette raison que je crée mon arbre dans le server, puis je l'envoie créé et sérialisé au client. Le client va juste le désérialiser et l'ajouter à a son interface.

Dans listDir() de CommandDispatcher : *DefaultMutableTreeNode userRoot = client.getRootNode(); window.root.add(userRoot);*

Change working directory

Pour cela j'ai rajouté un bouton btnCwd à l'interface car c'est plus intuitif pour un utilisateur. Un textField augmente la marge d'erreur car elle implique que l'utilisateur rentre exactement le bon nom du dossier.

Afin d'empêcher les mauvaises manœuvre je vérifie si c'est bien dans un dossier qu'on veut se déplacer avant de changer le répertoire courant.

Voir fonction cwd() dans CommandDispatcher.

Download a file

Dans cette partie aussi je laisse l'utilisateur choisir le fichier qu'il veut en cliquant dans l'arborescence.