



REPUBLIQUE DE COTE D'IVOIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE FELIX HOUPOUËT BOIGNY D'ABIDJAN COCODY



UFR : MATHEMATIQUES - INFORMATIQUE

ANNEE UNIVERSITAIRE : 2022-2023

UE : RESEAUX ET PROTOCOLES

NIVEAU : MASTER 1 BDGL

RAPPORT DOCUMENTANT NOTRE IMPLEMENTATION

NOM ET PRÉNOMS DES PARTICIPANTS :

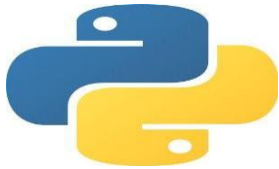
KOUASSI KONAN FERNAND

N'GUETTIA KOUAKOU FRANCE LEONNEL

NOM DU PROF :

M. DIALLO MOHAMED

Transfert de fichiers Multithread en utilisant le Socket TCP en Python

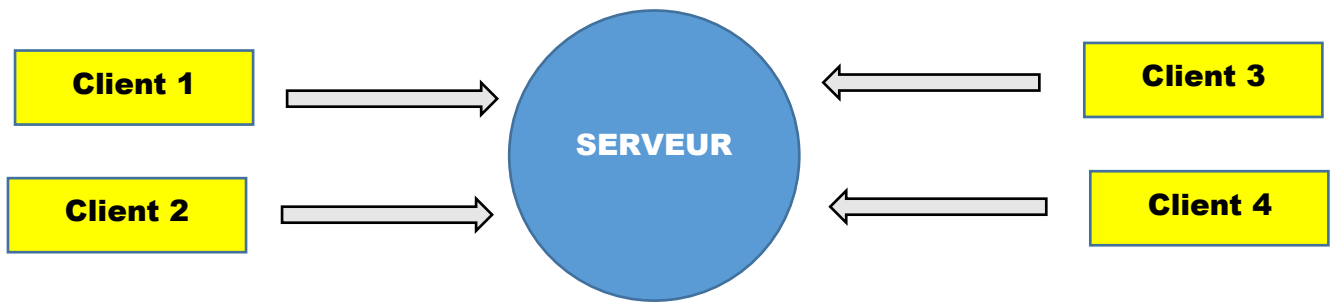


INTRODUCTION

Dans le cadre de ce projet, nous avons développé un serveur de fichiers et un protocole de transfert de fichiers en équipe de deux. Le but de ce projet est de permettre à des clients d'afficher la liste des fichiers, de télécharger des fichiers à partir d'un répertoire spécifié dans la configuration du serveur.

Dans ce rapport, nous documenterons l'implémentation de notre solution en utilisant le langage python.

Architecture de la solution



Notre solution est basée sur une architecture client-serveur, où le serveur héberge les fichiers et les clients peuvent se connecter au serveur pour télécharger les fichiers. Nous avons utilisé le protocole de communication TCP pour la communication entre le client et le serveur.

Le serveur est capable de gérer plusieurs connexions clientes simultanément. Chaque client peut demander la liste des fichiers hébergés sur le serveur et télécharger un ou plusieurs fichiers de cette liste avant de terminer la session.

Fonctions

1. **LIST** : renvoie la liste des fichiers stockés dans le dossier `server_data` au client.
2. **DOWNLOAD** : demande au serveur de télécharger un fichier et de le stocker dans le dossier `client_data`.
3. **HELP** : envoie une liste des commandes disponibles au client.
4. **LOGOUT** : se déconnecte du serveur.

Implémentation

Nous avons implémenté la solution en utilisant le langage Python. Le code est organisé en plusieurs fichiers qui contiennent différentes parties de l'application.

Le fichier `'serveur.py'` contient le code pour le serveur. Il utilise la bibliothèque standard de Python pour les sockets pour écouter les demandes des clients et gérer les connexions. Le serveur utilise également un objet `'File Handler'` pour gérer les demandes de fichiers des clients.

Le fichier `'client.py'` contient le code pour le client. Il utilise la bibliothèque standard de Python pour les sockets pour se connecter au serveur et envoyer les demandes.

Protocole de transfert de fichiers

Le protocole de transfert de fichiers que nous avons implémenté fonctionne comme suit :

1. Le client se connecte au serveur en utilisant une socket TCP.
2. Le client envoie une demande de liste de fichiers au serveur en envoyant le message "LIST".
3. Le serveur renvoie la liste des fichiers hébergés sur le serveur au client.
4. Le client envoie une demande de téléchargement de fichier au serveur en envoyant le message "DOWNLOAD filename".
5. Le serveur envoie le fichier demandé au client.
6. Le client peut répéter les étapes 4 et 5 pour télécharger d'autres fichiers.
7. Le client envoie le message "LOGOUT" pour terminer la session.
8. Le serveur ferme la connexion avec le client.

CONCLUSION

Dans ce rapport, nous avons documenté l'implémentation de notre solution pour le protocole de transfert de fichiers en utilisant Python. Notre solution permet aux clients de télécharger des fichiers à partir d'un répertoire spécifié dans la configuration du serveur.

Nous avons utilisé une architecture client-serveur avec le protocole de communication TCP pour la communication entre le client et le serveur.