

## UTILISATION

### COMPILATION :

gcc -o server server.c

gcc -o client client.c

### UTILISATION :

./client <IP> <PORT>

./server <IP> <PORT>

## FONCTIONNEMENT GLOBAL

### SERVEUR :

Exécution du script avec les arguments nécessaires

Création du socket grâce aux arguments renseignés par l'utilisateur

Bind et écoute sur le port et l'IP passés en arguments

Acceptation de la connexion

Entrée dans une boucle (1)

Réception de la commande

Exécution de la commande sur le serveur

Entrée dans une boucle (2) qui récupère le résultat des commandes ligne par ligne

Récupération du résultat et stockage

Envoi du résultat au client

Envoi de la chaîne de fin de résultat

Fermeture de la boucle (2)

Fermeture de la boucle (1)

Fermeture du socket

### CLIENT :

Exécution du script avec les arguments nécessaires

Création du socket grâce aux arguments renseignés par l'utilisateur

Tentative de connexion au serveur

Entrée dans une boucle (3)

Entrée d'une commande par l'utilisateur

Récupération de l'entrée et stockage

Comparaison de l'entrée avec la commande sortie

Envoi de la commande au serveur

Entrée dans une boucle (4) qui reçoit la réponse du serveur ligne par ligne

Vérification que la ligne n'est pas la ligne de fin

Affichage de la réponse à l'utilisateur

Réception de la chaîne de fin de résultat

Fermeture de la boucle (4)

Nouvelle entrée utilisateur (boucle (3))

Si l'entrée est exit fermeture de la boucle (3)

Fermeture du socket

## CHOIX D'IMPLÉMENTATION

- TCP : J'ai opté pour TCP plutôt que UDP pour sa fiabilité. Les commandes ainsi que leurs résultats ne seront pas altérés lors des échanges.
- fgets : J'ai choisi fgets plutôt que fgetc pour permettre la récupération ligne par ligne du résultat et donc permettre la gestion directe d'une grande quantité de lignes.
- Dû à la sortie de certaines commandes, on s'assure d'avoir reçu tout le contenu en ajoutant un message de fin de transaction qui va permettre de clôturer la boucle en cours

## GITHUB

<https://github.com/etu43598/devprojectrevshell>