

Dans ce TP nous allons implémenter un petit serveur de fichier au travers d'une connexion TCP et implémentée en C. Le TP doit être réalisé sur un system UNIX (Linux ou MacOS).

Le fonctionnement du serveur est le suivant :

1. Création d'un socket qui écoute (listen) sur le port 2020 en IPv4
2. Attente d'une nouvelle connexion, stocke le nouveau socket dans une variable `socketcli`.
3. Lis les données reçues sur la connexion jusqu'à un saut de ligne `\n`. Les données reçues sont stockées dans une chaîne de caractère `path`. Les données reçues après le saut de ligne sont ignorées.
4. Ouvre le fichier dont le nom est `path`
5. Écrit le contenu du fichier dans `socketcli`.
6. Ferme le fichier, ferme la connexion
7. Retourne à l'étape 2.

On pourra supposé que la taille maximale des chemins est de 512 octets.

Question 1 Créé un programme C qui réalise l'étape 1. Tester. *On pourra utiliser l'exemple du cours*

Question 2 Réalisez l'étape 2, exécutez votre programme, il doit attendre une nouvelle connexion. Faites une connexion dans une autre console avec la commande `nc 127.0.0.1 2020`, votre programme doit s'arrêter sans erreur.

Rappel de chaîne de caractère en C En C il existe deux manières d'encodées les chaînes de caractère, les deux ont le type `char *` ou `char []` la différence réside dans l'encodage de la taille de la chaîne :

- Les chaînes de caractère terminée par `null` qui ne peuvent pas contenir de donnée arbitraire, car elle pourrait contenir `null` ;
- Les tableaux de caractère + la taille de la chaîne. C'est ce qui est utilisé pour les appels système `read` et `write` pour pouvoir écrire des données arbitraires.

Pour passer de la seconde représentation à la première, il faut ajouter un caractère `null` à la fin de la chaîne.

Question 3 Réaliser l'étape 3, commencé par lire toutes les données dans le socket pour les mettre dans un buffer. Chercher le saut de ligne. *Vous pouvez utiliser la fonction `index` de la librairie standard. Faites attention au fait que votre buffer ne contient pas le caractère `null` à la fin des données. (c.f. rappel ci dessus ; Me poser la question si vous ne comprenez pas !)*

Question 4 Tester l'étape précédente en modifiant le code pour imprimer la chaîne de caractère lue. Vous pouvez utiliser la commande `echo "test\n" | nc localhost 2020` pour ouvrir une connexion et écrire la chaîne de caractère "test".

Question 5 Réaliser les étapes 4 et 5. *L'ouverture d'un fichier se fait avec `open` dont vous pouvez chercher la description dans le manuel.* La lecture et l'écriture d'un fichier se font de la même manière qu'avec un socket.

Question 6 Réaliser les étapes 6 et 7 et tester.

Question 7 Faire la même chose en UDP.