Programmation Unix
TP
Client/Serveur en mode connecté

Le but de ce TP est d'écrire un programme client (nous le nommerons strmcli) et un programme serveur (nous le nommerons strmser). La communication entre le serveur et le client se fera en mode connecté.

Question 1 Client/Serveur séquentiel

Description

Le serveur est de type séquentiel et effectue une boucle dans laquelle il affiche le message :

ATTENTE D UNE CONNEXION.....

Puis, il passe en attente d'une demande de connexion de la part d'un client.
Lorsqu'un client se connecte, il affiche le message :

JE TRAITE LA CONNEXION QUI VIENT D ARRIVER….

Le traitement consiste à afficher:

- Le nom et l'adresse IP de la machine du client
- Le n° de port utilisé par le client
- Le nombre de caractères envoyés par le client
- La chaîne de caractères qui lui est envoyée par le client.

Le serveur se termine lorsqu'un client lui envoie une chaîne de caractères composée du seul caractère #. Il affiche alors le message : *FIN DE SERVICE*

Description du client

Le client invite l'utilisateur à entrer une chaîne de caractères. Il se connecte ensuite au serveur puis lui envoie la chaîne de caractères entrée par l'utilisateur. Enfin, il se termine.

Adresse du serveur

Le serveur adoptera l'adresse de la machine sur laquelle il fonctionne. Cette adresse sera déterminée à l'aide de `gethostname` et `gethostbyname`.

N° de port du serveur

Le serveur n'étant pas répertorié dans l'annuaire, il est inutile d'utiliser `getservbyname` pour déterminer son numéro de port. Chaque étudiant fera tourner son serveur sur sa machine de travail au moment du TP. Le numéro de port de son serveur sera déterminé de la façon suivante : 5600 + N° de son PC.

Par exemple, si le PC est pc508-12, le n° de port du serveur sera 5600 + 12 = **5612**. De cette manière, tous les serveurs développés auront un n° de port différent au moment du TP.

Attention, si ce TP s'étale sur plusieurs séances et que vous changez de PC, ce n° de port devra être recalculé en conséquence.

Lancement du serveur

```
strmser Numero_port_serveur
```

Localisation du client

Chaque étudiant fera tourner son client sur une machine différente de la machine du serveur.

Lancement du client

```
strmcli Nom_machine_serveur Numero_port_serveur
```

NB :

- Les variables de type `struct sockaddr_in` seront mises à zéro avec `bzero` avant toute utilisation.
- Le champ `sin_addr` sera valué en utilisant `bcopy`.
- Le champ `sin_port` sera valué en utilisant `htons`.
- Tous les codes retour des fonctions de l'API socket seront testés et, en cas d'erreur, un message sera affiché au moyen de `perror` (cf. man).
- Vous vous préoccuperez également des valeurs NULL éventuellement retournées par `gethostbyname` qui seront signalées par un message (`printf`).

Question 2 Client/Serveur concurrent

Si vous le souhaitez, réécrire le serveur de façon concurrente.

Question 3 Client/Serveur multi-threadé

En utilisant les threads, réécrire le serveur afin que :

- il réponde sur plusieurs sockets d'écoute à la fois
- il propose à l'utilisateur un menu pour:
    - afficher la liste des connexions en cours
    - arrêter le serveur