

Modèle client/serveur en mode non connecté : un exemple dans le domaine AF_UNIX

Exercice 1 *Modèle client/serveur* Dans le modèle client/serveur le plus courant, le client fait une requête auprès d'un serveur qui lui envoie une réponse.

- 1) Comment le client connaît-il l'adresse du serveur ?
- 2) Comment le serveur peut-il répondre au client ?
- 3) Le client a-t-il besoin d'une adresse externe ?

Exercice 2 *Appels systèmes de manipulation de points de communication réseau*

- 1) Qu'est-ce qu'un point de communication réseau ?
- 2) Comment créer un point de communication dans l'interface des sockets de Berkeley ?
- 3) Comment recevoir des données sur un point de communication réseau ?

Exercice 3 *client/serveur echo* Le but de ce TD/TP est d'écrire un client/serveur écho en mode non connecté. Ce serveur renvoie ce que le client lui a écrit (principe de l'écho).

- 1) Décrire le modèle de ce serveur (schéma général).
- 2) Donner les primitives C permettant l'écriture d'un tel serveur en manipulant des sockets du domaine AF_UNIX.
- 3) Décrire un client permettant de se connecter à ce serveur et donner les primitives associées.

Exercice 4 *TP*

- 1) (TP) Écrire en C un serveur echo en mode non connecté dans le domaine AF_UNIX. Dans le cas où le client envoie une requête, le serveur lit les caractères envoyés par le client et les lui retourne.
- 2) (TP) Écrire en C un client pour le serveur de la question précédente. Celui-ci envoie ligne par ligne les messages reçus depuis l'entrée standard. Chaque fois qu'une ligne est lue sur l'entrée standard, elle est envoyée vers le serveur. Le client attend la réponse du serveur, affiche celle-ci à l'écran et attend une nouvelle ligne sur l'entrée standard et ceci jusqu'à fin de fichier.
- 3) (TP) Veiller à ce que votre serveur s'arrête correctement sur un signal de votre choix, et ignore les autres signaux d'interruption.
- 4) (TP) Adapter votre serveur pour qu'il puisse gérer plusieurs clients simultanément. Pensez à gérer les processus zombies.
- 5) (TP) Ajouter une fonctionnalité de log pour votre serveur : à chaque demande de client, le serveur conservera dans un fichier l'adresse du client, la date et la question du client.
- 6) (TP, optionnel) Pouvez-vous déterminer la taille maximale des données pouvant être envoyées en une fois par votre client ?