

Sockets Unix en mode connecté

Ce TD/TP a pour objectif la compréhension des appels système utilisés pour la communication socket en mode connecté, de nouveau dans le domaine `AF_UNIX`. Vous devrez rendre sur Universitice un compte-rendu lié aux questions de TP ; il doit être au format PDF, soit à partir d'un code LaTeX, soit à partir d'un logiciel de traitement de texte type LibreOffice. Les réponses doivent être rédigées et quand c'est possible illustrées par des captures d'écran de votre travail (ligne de commande, Wireshark, etc.). Vous soumettrez aussi vos captures (dans la mesure du raisonnable concernant leur taille) ainsi que vos éventuels fichiers C.

Exercice 1 *Appels systèmes de manipulation de points de communication réseau en mode connecté*

- 1) Rappeler comment un processus peut créer un point de communication réseau avec une adresse externe, en précisant les différences entre le mode connecté et le mode non connecté.
- 2) Comment le serveur déclare-t-il au noyau qu'il est en écoute (en attente de connexions) sur ce point de communication ?
- 3) Comment le client fait-il pour demander une connexion au serveur ? Comment le serveur fait-il pour accepter une connexion d'un client ?
- 4) Expliciter ce que sont une socket d'écoute, une socket de service.
- 5) Dans le mode connecté, en `AF_UNIX`, le client a-t-il besoin d'une adresse externe ? Comment le serveur fait-il pour lui envoyer une réponse ?

Exercice 2 *Modèle client/serveur mode connecté*

- 1) Décrire le modèle général d'un serveur en mode connecté.
- 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 3 *TP*

- 1) (TP) Écrire un client et un serveur (domaine `AF_UNIX`, mode connecté) permettant de jouer à « trouver un nombre entre 1 et 1000 ». Chaque joueur sera pris en charge par un thread distinct et jouera indépendamment des autres joueurs.
- 2) (TP) Veiller à ce que votre serveur s'arrête correctement sur un signal de votre choix, et ignore les autres signaux d'interruption.
- 3) (TP) Le modifier pour que plusieurs joueurs puissent proposer simultanément un nombre.