TD Programmation Système

Série 7 : Communication Interprocessus par Socket

Exercice 1 : Test de socket

1. Ecrire le programme C sous Unix qui permet à un processus de créer une socket de domaine UNIX en mode non connecté et qui affiche son descripteur.

   Appel Système : socket(), perror()

2. Ecrire le programme C sous Unix qui permet à un processus de créer une socket de domaine UNIX en mode non connecté et qui attache la socket à une adresse de transport sous forme d'un fichier dont le chemin d'accès est une chaîne de caractère passé en paramètre.
   Appel Système : socket(), bind(), perror()

3. La commande netstat permet de connaitre l'état du réseau, les sockets crées, leurs états…

   a. Exécuté la commande netstat
   b. Modifie le programme de la question 2 pour pouvoir observer par la commande netstat la socket crée en mode non connecté et son attachement au fichier passé en paramètre

Exercice 2 : communication interprocessus par socket

1. Communication en mode UDP unidirectionnelle

Deux processus P1 et P2 communiquent en mode non connecté en communication Locale. Le processus P1 récupère un caractère l'envoi au processus P2. Ce dernier transforme le caractère reçu en majuscule et il l'affiche.

Ecrire en C les programmes P1 et P2 en utilisant les sockets sous Unix.

Appel Système : socket(), bind(), sendto(), recvfrom(), perror()

2. Communication en mode UDP bidirectionnelle

Modifier les programmes P1 et P2 pour que le caractère reçu et transformé en majuscule par P2 soit renvoyé au processus P1 qui l'affiche.