

Programmation réseau avec sockets en C sous Linux (2)

V. FELEA & A. HUGEAT & E. MERLET

1. Statistiques réseau - netstat

Les activités réseau de la machine peuvent être surveillées grâce à la commande Linux/Windows **netstat**.

Les réponses aux questions suivantes sont attendues pour le système Linux.

- Q1 Quelle option de quelle commande système permet d'afficher les processus de la machine utilisant le protocole IPv6 ? Donner la commande système complète. Quel est le résultat de la commande ?
- Q2 Donner la commande système qui permet d'afficher exclusivement la liste des adresses IP de la machine courante associées à des sockets connectées. Quel est le résultat de son exécution ? Mêmes questions pour les adresses IP des machines distantes ayant une connexion par sockets à la machine courante. Donner la commande permettant d'afficher les adresses IP de la machine courante associées à des sockets en mode non connecté.
- Q3 Afficher les sockets en état LISTEN. La commande système doit être fournie.
- Q4 Donner la commande système permettant d'afficher les ports utilisés par les sockets locales en mode connecté. Quels sont les processus qui les utilisent ?
- Q5 Rajouter des temporisations dans les programmes de l'application d'envoi de message et identifier les différents états des sockets utilisées. Expliquer ces états.
- Q6 Dans la version avec temporisations, une fois la connexion établie entre le client et le serveur, fermer l'application serveur "brutalement" (Ctrl-C) et essayer de la relancer immédiatement. Qu'obtenez-vous ? Qu'indique **netstat** ? Expliquer, en utilisant les phases de la fermeture d'une connexion TCP.

Réutilisation d'adresse Utiliser la configuration suivante sur la socket de connexion ouverte sur le serveur, après sa création et avant l'appel **bind**.

```
int enable = 1;
if (setsockopt(sockfd, SOL_SOCKET, SO_REUSEADDR,
               &enable, sizeof(int)) < 0)
    /* error */
```

Refaire le test précédent. En conclure.

2. Échanges en mode connecté (utilisant le protocole TCP)

2.1 Dire Bonjour (sens bidirectionnel de communication)

Modifier l'application du TP précédent telle qu'une application envoie un nom et l'autre lui répond par "Bonjour *nom*". La taille maximale du nom est connue par les deux applications (donnée par une constante).

2.2 Carré d'un nombre entier

Écrire une application client/serveur où le client sollicite au serveur le calcul du carré d'un entier qui lui transmet. Le client affichera la valeur reçue.