

PR6 – Programmation réseaux

TP n° 10 : Entrées et sorties non-bloquantes

Exercice 1 : Utiliser `select` pour un serveur echo

Écrivez un serveur TCP qui écoute sur un port et agit en echo pour retourner les messages reçus (une fois un client connecté sur ce port, si un message arrive du client, il lui renvoie). Les messages auront taille maximale 512 octets et termineront par `\n`. Ce serveur doit :

- permettre la connexion et l'echo de plusieurs client en parallèle
- ne pas utiliser des threads mais utiliser `select`

Le serveur devra donc créer et utiliser deux `fd_set` :

- un `fd_set` `a_surveiller`
 - il contiendra au départ le file descriptor correspondant au socket du serveur (pour surveiller les nouvelles connexions)
 - on lui ajoutera à chaque nouvelle connexion les file descriptor des sockets clients
- un `fd_set` `active`
 - au démarrage de la boucle recevra une copie de `a_surveiller`
 - sera mis à jour par `select` pour détecter les nouveaux événements sur les sockets clients et faire l'echo

On testera le serveur avec `netcat` sur le port choisi.

Exercice 2 : Echo avec `poll`

Faites la même chose qu'à l'exercice 1 mais en utilisant `poll` au lieu de `select`. Il faudra cette fois maintenir un tableau de `struct pollfd` au lieu de la liste `fd_set` de l'exercice précédent.

Exercice 3 : Du Calcul avec `timeout`

Écrivez un serveur qui attend une connexion d'un client puis fait se dérouler le jeu suivant : il lui envoie un message (qui termine par `\n`) contenant une multiplication au hasard à calculer (on pourra se restreindre par exemple à des nombres entre 1 et 20) et le joueur coté client a 10 secondes pour répondre. Le serveur enverra donc un décompte `10,9,8,...` qui sera affiché dans le terminal du client. Si le joueur envoie la bonne réponse avant la fin des 10 secondes alors le serveur compte un point et lui en envoie une autre. Sinon il envoie le message "TROP LENT!" puis envoie ensuite une nouvelle multiplication à calculer. Au bout de 5 questions le programme s'arrête et le serveur envoie au joueur le nombre de bonnes réponses obtenues. Là encore on pourra tester le serveur avec `netcat` sur le port choisi.