

TP AISE — Sockets

Exercice 1 : Client TCP

Réalisez un programme qui se connecte à `towel.blinkenlights.nl` sur le port 23 et affiche le contenu renvoyé sur la sortie standard que voyez vous ?

Exercice 2 : Serveur TCP

Réalisez un serveur TCP simple qui répète ce qu'il reçoit en utilisant les trois techniques de gestion des clients multiples suivantes:

- Fork sur accept
- Création de multiple threads
- Multiplexage des descripteurs de fichiers

On pourra tester ce programme en envoyant de donnée avec le client de l'exercice 1 ou bien en reposant sur la commande « nc » (netcat) comme illustré en cours.

Exercice 3: Serveur WEB

En utilisant le cours, et en analysant le contenu d'une requête telle que générée par votre client web, et en expérimentant sur des serveur existant implémentez un serveur web simple qui supportera l'envoi de multiple pages (gestion de la directive GET). En particulier vous devrez pouvoir envoyer une page et suivre un lien sur cette même page. Les page étant des fichiers locaux. Dans le dépôt nous avons placé un page `index.html` à envoyer par défaut (/) et une seconde page « `subpage.html` » référencée par la première, elles peuvent servir d'exemple.

Exercice 4 : Serveur et Client de Messagerie

Vous implémenterez un serveur de chat simple avec votre propre protocole. Le but est que plusieurs clients, chacun avec un login puissent être dans la même salle de discussion et échanger au format texte. Le serveur doit gérer de multiples clients et permettre leur déconnexion et reconnexion. Les différents serveurs précédemment implémentés peuvent servir de base.

En complément, vous permettrez:

- Une commande de message privé sur le canal afin de vous permettre de parler à une seule autre personne, par exemple « `/mp toto XXXX` »
- Une commande pour lister les personne présentes, par exemple « `/list` »
- Une command pour kicker un utilisateur « `/kick toto` »