TP sockets : exercice n°1

- Récupérer les fichiers C sous Moodle
  - Fichier « tcp_serveur.c » : serveur TCP ;
  - Fichier « tcp_client.c » : client TCP ;
  - Makefile (pour compiler automatiquement).
- Compiler avec make.
- Exécuter les programmes :
  - Comment faut-il les exécuter ? Avec la commande ./Nom_fichier
  - Peut-on exécuter plusieurs clients ? Non car dans notre programme nous sommes limité à un client dans notre programme mais en général on peut.
  - Peut-on exécuter plusieurs serveurs ? Non

TP sockets : exercice n°2

- Copier les fichiers précédents dans un autre répertoire.
- Modifier les deux fichiers de manière à ce que :
  - Une fois un client connecté, le serveur envoie « Bonjour ! » au client (le client affichera ce qu'il a reçu) ;
  - Une fois le message reçu, le client enverra « Merci ! » au serveur (le serveur affichera ce qu'il a reçu) .
- Tester indépendamment :
  - Son serveur avec le client du groupe d'à côté ;
  - Son client avec le serveur du groupe d'à côté. ;
  - Que faut-il modifier ? Il faut modifier l'adresse ip du client pour mettre celle de la machine du groupe d'à côté

TP sockets : exercice n°3

- Copier les fichiers précédents dans un autre répertoire (revenir à des sockets en local).
- Modifier les deux fichiers de manière à ce que :
  - Le serveur puisse accepter en boucle des connexions.
  - Quel problème entrevoit-on ?

Comme on l'a dit précédemment on ne peut exécuter qu'un client à la fois : le client doit être fini avant qu'un autre puisse s'exécuter

TP sockets : exercice n°4

- Copier les fichiers précédents dans un autre répertoire.
- Modifier les deux fichiers de manière à ce que :
  - Le serveur puisse accepter en boucle des connexions mais crée un nouveau processus fils (« fork ») pour gérer la connexion avec le client, le processus père continuant d'attendre d'autres connexions clients ;
  - Le client demande la saisie d'une chaîne de caractères à l'utilisateur (utiliser la fonction « fgets »), l'envoie au serveur (qui affichera ce qu'il a reçu), puis sorte.

TP sockets : exercice n°5

- Copier les fichiers précédents dans un autre répertoire.
- Modifier les deux fichiers de manière à ce que :
  - On ait le même comportement que précédemment sauf que le client demande en boucle (infinie) des messages qu'il envoie au serveur.
  - On rattrape le Ctrl-C dans le client de manière à ce que la socket soit convenablement fermée avant de quitter.

TP sockets : exercice n°6

- Copier les fichiers précédents dans un autre répertoire.
- Modifier les deux fichiers de manière à ce que :
  - On utilise une « thread » plutôt qu'un « fork ». Qu'est-ce qui change ? À quoi doit-on faire attention ?

On gère ici plusieurs connexions en parallèles. Les threads ont des variables partagées mais ici l'enjeu et que chacun ait des « variables locales » pour gérer sa propre connexion.