Perl
Processus et API sockets

Exercice 1 : Sockets

Q 1. Écrire deux scripts :
– un serveur, nommé `simple-serveur`, à un argument. L'argument est un nombre qui désigne le nombre de fois que le serveur envoie le mot `Bonjour` à chacun de ses clients ;
– un client, nommé `simple-client`, qui se connecte au serveur et affiche tout ce qu'il reçoit.
Essayer de lancer plusieurs clients.

Q 2. Écrire un client `http` en texte, nommé `http-client`. L'utilisateur passe en argument le nom de la machine serveur et le nom du fichier (html) qu'il désire charger, le client affiche à l'écran le fichier.

Pour mémoire la RFC 2616 définit le protocole HTTP version 1.1 et notamment la requête de demande d'un fichier par l'envoi sur le port 80 du serveur d'un texte similaire à l'exemple ci dessous :

Requête HTTP

```
1 GET /emploi_du_temps.html HTTP/1.1\n
2 Host: da2i.univ-lille1.fr\n
3 \n
```

Q 3. Écrire un client du service `echo`, nommé `echo-client`. Ce service est très simple : le serveur attend sur le port `7` d'une machine et renvoit au client tout ce que le client lui envoie, *il repète*. Utilisez votre machine local comme serveur pour vos tests.

Q 4. Écrire un serveur du service `echo`, nommé `echo-serveur`. Vous ne pouvez pas lancer votre serveur sur le port `7`, nous choisirons donc `7777`.

Exercice 2 : Processus

Q 1. Écrire un script, nommé `rien`, qui :

1. affiche un message contenant son `pid` indiquant qu'il démarre,
2. attend 3 secondes,
3. affiche un message contenant son `pid` indiquant qu'il termine.

Q 2. Écrire un script, nommé `n-fois-rien`, qui prend en argument un entier n et qui crée n fils qui exécutent le script `rien`. Chaque fois qu'il a créé un fils, le père affiche un message indiquant la création et le `pid` du fils créé. Ensuite le père doit attendre la terminaison de ses fils **dans leur ordre de création** et afficher un message chaque fois qu'il s'aperçoit qu'un de ses fils s'est terminé.

Q 3. Écrire un script, nommé `n-fois-rien-au-plus`, qui prend en argument un entier n, qui représente le nombre maximum de fils que le processus peut avoir à un instant donné. Chaque fois que l'utilisateur saisit quelque chose au clavier, le père essaie de créer un fils qui exécute le script `rien`. Si la création réussit, le père affiche le `pid` du fils qu'il vient de créer. Si le nombre maximum de fils possible est atteint, le père affiche qu'il doit attendre, attend et, dès qu'un fils se termine, affiche le `pid` du fils terminé et lance le fils suivant.

Q 4. Modifier votre serveur `echo`, en le nommant `echo-serveur-2`, afin qu'il puisse traiter plusieurs clients en **même temps**.