UNIVERSITÉ CAEN NORMANDIE

TD/TP : *sockets* locales & réseau

16 novembre 2018

Serveurs Itératif versus Parallèle

---

1 Introduction

Dans ce TD/TP nous allons voir qu'il est très facile de passer d'un code permettant de faire une communication locale à un code permettant une communication réseau, voir comment passer d'un serveur Itératif à un serveur parallèle .

Nous allons aussi voir l'intérêt de la primitive ***htons();***

Pour ce TP, vous devez travailler avec une ou plusiseurs VM à l'aide de Virtualbox. ! ! !

Vos VM vont avoir 2 interfaces réseaux Eth0 et Eth1 comme le montre l'image suivante (Merci à Vincent ☺ !)

```
root@2017-09-22-strech-lvm:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:75:fd:2e brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe75:fd2e/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:fc:44:c0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fefc:44c0/64 scope link
       valid_lft forever preferred_lft forever
root@2017-09-22-strech-lvm:~# ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.588 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.717 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.751 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.666 ms
^C
--- 192.168.56.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3029ms
rtt min/avg/max/mdev = 0.588/0.680/0.751/0.066 ms
root@2017-09-22-strech-lvm:~#
```

2 Exercice 1 : en local

L'objectif de ce premier exercice est de comprendre les mécanismes de base des *sockets.*

1. **Saisissez** les 2 codes ***serveur_local.c*** et ***client_local.c*** du chapitre **La communication en local** du cours et les **compilez** les... Par manque de temps **récupérez-les** sur ecampus
2. **Lancez** en 1er le client et **interprétez** le message d'erreur.
3. **Lancez** le serveur en tâche de fond (**&**) et **vérifiez** qu'il "tourne".
4. **Lancez** la commande ***netstat -l*** et **vérifiez** qu'il y a bien un serveur qui écoute (***listen***).
5. **Lancez** la commande ***ls -rtl*** et **vérifiez** qu'il y a bien un fichier de type *socket socket_serveur.* Expliquez alors pourquoi je dis parfois qu'une socket est un fichier
6. **Lancez** un ou plusieurs clients (***./client_local & ./client_local & ./client_local & ./client_local***).
7. **Interprétez** et **Concluez**.

```
root@debian95-Rx-Sys-Fougeray:~/L3/TD_TP/socket/local# ./serveur_local &
[1] 1072
root@debian95-Rx-Sys-Fougeray:~/L3/TD_TP/socket/local# Le serveur attend

root@debian95-Rx-Sys-Fougeray:~/L3/TD_TP/socket/local# ./client_local
Le client a écrit : A
Le serveur a lu : A
Le serveur écrit : B
Le serveur attend
Le serveur a renvoyé : B
root@debian95-Rx-Sys-Fougeray:~/L3/TD_TP/socket/local#
```

3 Exercice 2 : en réseau...

1. **Expliquez** rapidement la différence entre une communication dans le domaine AF_UNIX et une communication dans le domaine AF_INET sur le protocole TCP/IP donc en mode connecté.
   Par manque de temps **récupérez-les** sur ecampus
2. **Modifiez** (sauf si vous les avez récupérés sur ecampus) les 2 codes précédents afin de réaliser un *serveur_reseau* et un *client_réseau* qui ne vont accéder qu'à *localhost* sur le port 1664. Pour cela vous devez modifier simplement :
   — Le type de socket,
   — Le nommage du socket,
   — type,
   — adresse IP,
   — Nř de port autorisé, par exemple 1664 :) pourquoi pas 33 ?
   En vous inspirant du code ***créer_socket.c*** de la page 2 du cours.
3. **Lancez** la commande ***netstat -ltp*** et **vérifiez** qu'il y a bien un serveur qui écoute (***listen***)
4. **Lancez** 3 *clients :* **./client_reseau & ./client_reseau & ./client_reseau**
5. **Interprétez** et **Concluez**.
6. **Modifiez** dans le code source du *serveur_reseau* :
   la ligne ***adresse_serveur.sin_port=htons(1664);*** en ôtant la primitive ***htons***
   recompilez le serveur.
7. **Relancez** le serveur et **refaites** ***netstat -lt*** et **vérifiez** qu'il y a bien un serveur qui écoute (***listen***)
8. **Relevez** le Nř de port, **comparez**-le à 1664, **vérifiez** que cette valeur 32774 est bonne...
9. **Concluez**.

```
root@debian95-Rx-Sys-Fougeray:~/L3/TD_TP/socket/en_reseau# ./serveur_reseau &
[4] 1283
[3]   Processus arrêté      ./serveur_reseau
root@debian95-Rx-Sys-Fougeray:~/L3/TD_TP/socket/en_reseau# Le serveur attend

root@debian95-Rx-Sys-Fougeray:~/L3/TD_TP/socket/en_reseau# ps
  PID TTY          TIME CMD
  989 pts/0    00:00:00 bash
 1283 pts/0    00:00:00 serveur_reseau
 1284 pts/0    00:00:00 ps
root@debian95-Rx-Sys-Fougeray:~/L3/TD_TP/socket/en_reseau# ./client_reseau
j'envoie un A à mon serveur
Le serveur attend
 A moi client : 1285, le serveur me donne : B
root@debian95-Rx-Sys-Fougeray:~/L3/TD_TP/socket/en_reseau#
```

4 Exercice 3 : Itératif versus Parallèle...

Le serveur que nous avons écrit était un serveur dit **itératif**, dans cet exercice nous allons modifier le code de façon qu'il devienne un serveur **parallèle**.

1. **Rappelez** la différence entre un serveur itératif et un serveur parallèle.
2. **Expliquez** comment un serveur parallèle fonctionne d'un pont de vue processus
3. **Modifiez** le code du serveur itératif de manière qu'il devienne un serveur parallèle.
   Vous pouvez pour cela vous aider de l'image de la page 5 du cours.
   Vous devez utiliser la primitive ***signal***, de manière que le père ignore la terminaison des fils.

4. **Utilisez** ce serveur et comparez avec le précédent.
   Vous ne constatez rien de différents ?
   Nous allons ajouter du code de manière à faire “travailler” le serveur.
5. **Ajoutez** la ligne de code suivante *sleep(10)* [1] *;* après la ligne ***ch++****;* pour les 2 types de serveurs.
6. **Compilez**-les et **Lancez** les.
7. **Lancez** 3 clients avec la commande : ***time(./client_reseau & ./client_reseau & ./client_reseau & ./client_reseau)***
8. **Comparez** les 2 résultats
9. **Concluez**.

5 Exercice 4 : Serveur parallèle

Dans cet exercice vous n’allez pas coder... c’est bien trop long... Vous allez juste lancer le script shell qui suit et que vous pouvez analyser.

1. **Récupérez** les 3 codes sur ecampus dans le répertoire Serveur_parallèle du répertoire socket
2. **Ouvrez** les 3 fichiers C et **analysez** les.
3. **Lancez** avec la commande : ***lance_client_reseau.sh | tee resultat_lance_client_reseau.sh.txt***
4. **Patientez** ça dure un certain temps ! ! ! Ceci est dû aux 2 lignes ***netstat -npt | grep 50000***

```
#script lance.bash dans /root/L3/TD_TP/socket/parallele
#/bin/bash
clear
gcc client_reseau.c -o client_reseau
gcc  serveur_reseau_parallele.c -o  serveur_reseau_parallele
killall serveur_reseau_parallele
netstat -npt | grep 50000
./serveur_reseau_parallele &
netstat -t | grep 50000
#On lance les 6 clients qui attendent 3 secondes apres connexion
./client_reseau &
./client_reseau &
./client_reseau &
./client_reseau &
./client_reseau &
./client_reseau &
ps -jH
netstat -npt | grep 50000
echo "Fini, au boulot on explique le resultat... ;-)"
```

Vous devez obtenir un résultat de ce type !

5. **Expliquez** le !

```
root@debian95-Rx-Sys-Fougeray:~/L3/TD_TP/socket/chapitre_14/local# ./lance_client_reseau.s
rm: impossible de supprimer 'apres_serveur_serveur.txt': Aucun fichier ou dossier de ce ty

Le serveur attend
et il est 06h 15min 53sec.
A enfin un client ;-) .
et il est 06h 16min 13sec.
A enfin un client ;-) .
et il est 06h 16min 13sec.
A enfin un client ;-) .
et il est 06h 16min 13sec.
A enfin un client ;-) .
et il est 06h 16min 13sec.
la socket service vaut : 3
A enfin un client ;-) .
et il est 06h 16min 13sec.
A enfin un client ;-) .
et il est 06h 16min 13sec.
la socket service vaut : 3
```

1. *Le serveur travaille comme un étudiant fatigué en amphi le vendredi matin :)*

```
la socket service vaut : 3
la socket service vaut : 3
la socket service vaut : 3
la socket service vaut : 3
  PID  PGID   SID TTY          TIME CMD
 4964  4964  4964 pts/3    00:00:00 bash
 5313  5313  4964 pts/3    00:00:00   bash
 5329  5313  4964 pts/3    00:00:00     serveur_reseau_
 5341  5313  4964 pts/3    00:00:00       serveur_reseau_
 5342  5313  4964 pts/3    00:00:00       serveur_reseau_
 5343  5313  4964 pts/3    00:00:00       serveur_reseau_
 5344  5313  4964 pts/3    00:00:00       serveur_reseau_
 5334  5313  4964 pts/3    00:00:00     client_reseau
 5335  5313  4964 pts/3    00:00:00     client_reseau
 5336  5313  4964 pts/3    00:00:00     client_reseau
 5337  5313  4964 pts/3    00:00:00     client_reseau
 5338  5313  4964 pts/3    00:00:00     client_reseau
 5339  5313  4964 pts/3    00:00:00     client_reseau
 5340  5313  4964 pts/3    00:00:00     ps

je suis un fils, mon pid est 5341 et celui de papa est : 5329, ma SOCKET est 4
 A moi client : 5338, le serveur me donne : B
je suis un fils, mon pid est 5342 et celui de papa est : 5329, ma SOCKET est 4
 A moi client : 5337, le serveur me donne : B
je suis un fils, mon pid est 5343 et celui de papa est : 5329, ma SOCKET est 4
 A moi client : 5339, le serveur me donne : B
je suis un fils, mon pid est 5344 et celui de papa est : 5329, ma SOCKET est 4
je suis un fils, mon pid est 5345 et celui de papa est : 5329, ma SOCKET est 4
 A moi client : 5334, le serveur me donne : B
 A moi client : 5336, le serveur me donne : B
je suis un fils, mon pid est 5346 et celui de papa est : 5329, ma SOCKET est 4
 A moi client : 5335, le serveur me donne : B
================================================
diff encore_apres_clients.txt apres_serveur.txt
1,12d0
< tcp        0      0 localhost:40320         localhost:50000         ESTABLISHED
< tcp        0      0 localhost:40314         localhost:50000         ESTABLISHED
< tcp        0      0 localhost:40318         localhost:50000         ESTABLISHED
< tcp        0      0 localhost:50000         localhost:40318         ESTABLISHED
< tcp        0      0 localhost:50000         localhost:40312         ESTABLISHED
< tcp        0      0 localhost:40322         localhost:50000         ESTABLISHED
< tcp        0      0 localhost:50000         localhost:40316         ESTABLISHED
< tcp        0      0 localhost:50000         localhost:40320         ESTABLISHED
< tcp        0      0 localhost:40312         localhost:50000         ESTABLISHED
< tcp        0      0 localhost:50000         localhost:40322         ESTABLISHED
< tcp        0      0 localhost:50000         localhost:40314         ESTABLISHED
< tcp        0      0 localhost:40316         localhost:50000         ESTABLISHED
Fini, au boulot on explique le resultat... ;-)
root@debian95-Rx-Sys-Fougeray:~/L3/TD_TP/socket/chapitre_14/local#
```

6 Concluez