

## Exercices d'informatique - Série n° 3

Cours 4INOC01

Série distribuée le 16.11.2017

### 1. Les sockets.

- (1) Exécuter les deux scripts donnés ci-dessous dans deux fenêtres différentes. Que se passe-t-il ?

---

```
1 import socket , sys
2 host='127.0.0.1'
3 port=5000
4 c=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
5 try:
6     c.bind((host,port))
7 except:
8     print "Connexion impossible"
9     sys.exit()
10 c.listen(5)
11 connexion, adresse=c.accept()
12 print "Connexion de IP=",adresse[0],", port=",adresse[1]
13 connexion.send("Envoyez votre message.")
14 m=connexion.recv(100)
15 print m
16 connexion.close()
```

---

```
1 import socket , sys
2 host='127.0.0.1'
3 port=5000
4 c=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
5 try:
6     c.connect((host,port))
7 except:
8     print "Connexion impossible"
9     sys.exit()
10 m=c.recv(100)
11 print m
12 t=raw_input()
13 c.send(t)
14 c.close()
```

---

- (2) Modifier les scripts pour pouvoir envoyer un message à un camarade. Il faut mettre l'adresse IP du serveur (premier script) dans les deux programmes (ligne 2). Pour connaître l'adresse IP d'un ordinateur, sous linux, il suffit de taper la commande `ifconfig` dans un terminal. Par ailleurs, pour connaître la liste des ports ouverts, il faut taper la commande `netstat -an --inet -p`. Il faut, le cas échéant, modifier la ligne 3 dans les deux programmes. La commande `iptables` permet d'ouvrir un port mais elle nécessite des droits d'administrateur. Pour plus d'informations concernant ces commandes linux, il faut taper la commande `man ifconfig` ou `man netstat` ou `man iptables`.

### 2. Conversation.

Rédiger deux scripts Python, nommés `serveur.py` et `client.py`, qui permettent à deux utilisateurs distants de communiquer.

Le script `serveur.py` est lancé sur une machine qui sert de serveur et qui est utilisé par une des deux personnes pour communiquer. L'autre utilisateur exécute le script `client.py` qui lui

permet simultanément de recevoir et d'envoyer des messages au serveur. S'il tape "stop", les deux programmes s'arrêtent.

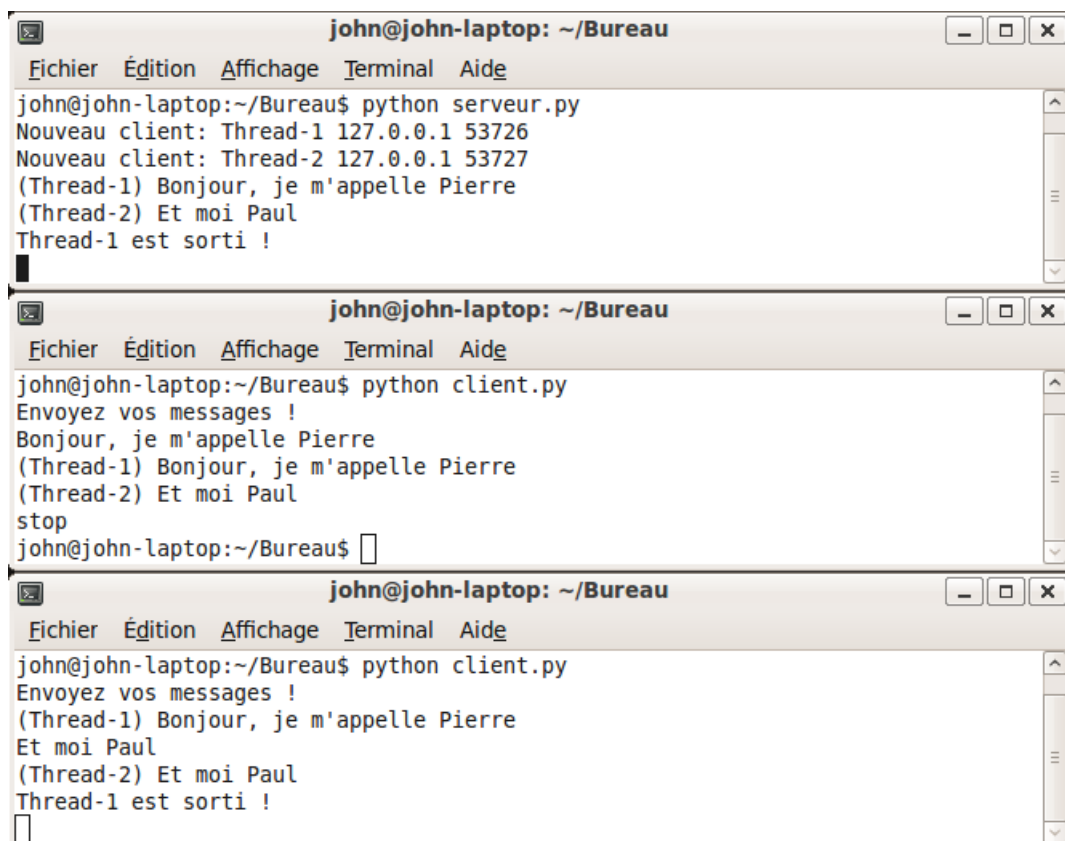
**Indications:** Le script `client.py` se compose de deux parties. Premièrement, il doit établir une connexion avec le serveur. Deuxièmement, il doit lancer deux processus parallèles. Le premier affiche les messages reçus et le deuxième gère l'envoi des messages au serveur. Ils doivent contenir chacun une boucle infinie. Le script `serveur.py` est similaire.

### 3. Conversations.

Rédiger deux scripts Python, nommés `serveur.py` et `client.py`, qui permettent de créer un réseau de communication entre différents ordinateurs. Une illustration du fonctionnement de ces scripts se trouve sur la figure 1.

Le script `serveur.py` est lancé sur une machine qui sert de serveur. Chaque utilisateur exécute le script `client.py` qui lui permet de recevoir et d'envoyer des messages au serveur. Le serveur transmet à tous les utilisateurs les messages qu'il reçoit.

**Indications:** Le script `client.py` se compose de deux parties. Premièrement, il doit établir une connexion avec le serveur. Deuxièmement, il doit lancer deux processus parallèles. Le premier affiche les messages reçus et le deuxième gère l'envoi des messages au serveur. Ils doivent contenir chacun une boucle infinie. Le script `serveur.py` doit contenir une boucle infinie qui crée un nouveau processus pour chaque nouvel utilisateur. Le processus réceptionne les messages de l'utilisateur et les transmet à tous les utilisateurs.



```
john@john-laptop: ~/Bureau
Fichier  Édition  Affichage  Terminal  Aide
john@john-laptop:~/Bureau$ python serveur.py
Nouveau client: Thread-1 127.0.0.1 53726
Nouveau client: Thread-2 127.0.0.1 53727
(Thread-1) Bonjour, je m'appelle Pierre
(Thread-2) Et moi Paul
Thread-1 est sorti !

john@john-laptop: ~/Bureau
Fichier  Édition  Affichage  Terminal  Aide
john@john-laptop:~/Bureau$ python client.py
Envoyez vos messages !
Bonjour, je m'appelle Pierre
(Thread-1) Bonjour, je m'appelle Pierre
(Thread-2) Et moi Paul
stop
john@john-laptop:~/Bureau$

john@john-laptop: ~/Bureau
Fichier  Édition  Affichage  Terminal  Aide
john@john-laptop:~/Bureau$ python client.py
Envoyez vos messages !
(Thread-1) Bonjour, je m'appelle Pierre
Et moi Paul
(Thread-2) Et moi Paul
Thread-1 est sorti !
```

FIGURE 1. Exercice 3