

Introduction à la programmation réseaux

Client de chat

```

1 import os, socket, sys
2
3 adresse_serveur = 'localhost'
4 numero_port = 6688
5 ma_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6 try: ma_socket.connect((adresse_serveur, numero_port))
7 except:
8     print ("probleme de connexion")
9     sys.exit(1)
10 pid = os.fork()
11 if (pid) :
12     while 1:
13         ligne = ma_socket.recv(1000)
14         print (str(ligne,encoding='UTF-8'))
15         if not ligne: break
16 else:
17     while 1:
18         clavier = input(':>')
19         if not clavier: break
20         ma_socket.sendall(bytes(clavier+'\n',encoding='UTF-8'))
21 ma_socket.close()

```

Serveur de chat

```

1 import sys, os, socket
2
3 adresse_hote = ''
4 numero_port = 6688
5 ma_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM, socket.IPPROTO_TCP)
6 ma_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
7 ma_socket.bind((adresse_hote, numero_port))
8 ma_socket.listen(socket.SOMAXCONN)
9
10 while 1:
11     (nouvelle_connexion, depuis) = ma_socket.accept()
12     print ("Nouvelle connexion depuis ", depuis)
13     nouvelle_connexion.sendall(b'Bienvenu\n')
14     pid = os.fork()
15     if (pid) :
16         while 1:
17             ligne = nouvelle_connexion.recv(1000)
18             print ("<:", str(ligne,encoding='UTF-8'))
19             if not ligne: break
20     else:
21         while 1:
22             clavier = input(':>')
23             if not clavier: break
24             nouvelle_connexion.sendall(bytes(clavier,encoding='UTF-8'))
25     nouvelle_connexion.close()
26 ma_socket.close()

```

Serveur de chat utilisant le Select

```
1 import sys, os, socket, select
2
3 adresse_hote = ''
4 numero_port = 6688
5
6 def lecture_ligne(ma_socket):
7     ligne = b''
8     while 1:
9         caractere_courant = ma_socket.recv(1)
10        if not caractere_courant :
11            break
12        if caractere_courant == b'\r':
13            caractere_suivant = ma_socket.recv(1)
14            if caractere_suivant == b'\n':
15                break
16            ligne += caractere_courant + caractere_suivant
17        continue
18        if caractere_courant == b'\n':
19            break
20        ligne += caractere_courant
21    return ligne
22
23 ma_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM, socket.IPPROTO_TCP)
24 ma_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
25 ma_socket.bind((adresse_hote, numero_port))
26 ma_socket.listen(socket.SOMAXCONN)
27
28 surveillance = [ma_socket]
29 while 1:
30     (evnt_entree, evnt_sortie, evnt_exception) = select.select(surveillance, [], [])
31     for un_evenement in evnt_entree:
32         if (un_evenement == ma_socket):
33             # il y a une demande de connexion
34             nouvelle_connexion, depuis = ma_socket.accept()
35             print ("Nouvelle connexion depuis ", depuis)
36             nouvelle_connexion.sendall(b'Bienvenu\n')
37             surveillance.append(nouvelle_connexion)
38             continue
39             # sinon cela concerne une socket connectée à un client
40             ligne = un_evenement.recv(1024)
41             if not ligne :
42                 surveillance.remove(un_evenement) # le client s'est déconnecté
43             else :
44                 print (un_evenement.getpeername(), ':', ligne)
45                 # envoyer la ligne a tous les clients, etc
46                 for desc in surveillance:
47                     if (desc != ma_socket) and (desc != un_evenement):
48                         desc.sendall(bytes(str(un_evenement.getpeername()), encoding='UTF-8')+b': '+ligne)
49
50 connexion.close()
51 ma_socket.close()
```