

UDP & Communication multicast

Serveur UDP

```
#!/usr/bin/python3

import socket, struct, time, sys, re, os

# configuration
mon_pseudo = b"prof"
numero_port = 7182
adresse_multicast = "224.0.0.127"
dico_pseudos = {}
fichier_pseudos = "les_pseudos.txt"

# programme
ma_socket_udp = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
TSAP = (adresse_multicast, numero_port)
ma_socket_udp.bind(('', numero_port))
mreq = struct.pack("4sl", socket.inet_aton(adresse_multicast), socket.INADDR_ANY)
ma_socket_udp.setsockopt(socket.IPPROTO_IP, socket.IP_ADD_MEMBERSHIP, mreq)

def diffuser_pseudo(une_socket):
    """Diffuse le pseudo"""
    while(1):
        une_socket.sendto(b"%s"%mon_pseudo, TSAP)
        time.sleep(10)

def sauver_liste_pseudos():
    try:
        fic = open(fichier_pseudos, 'bw')
    except Exception as e:
        print(e.args)
        sys.exit(1)
    for une_cle in dico_pseudos.keys():
        fic.write(b"%s:%s\n"%(une_cle, dico_pseudos[une_cle]))
    fic.close()

def lire_liste_pseudos():
    try:
        fic = open(fichier_pseudos, 'r')
    except Exception as e:
        print(e.args)
        return
    re_decompose = re.compile(rb'([^\:]+):(\d+\.\d+\.\d+\.\d+)')
    while(1):
        une_ligne = fic.readline()
        if not une_ligne: break
        resultat = re_decompose.search(une_ligne)
        if resultat:
            (pseudo, adresse_ip) = resultat.groups()
            dico_pseudos[pseudo] = adresse_ip
    fic.close()

def apprendre_pseudos_et_afficher_message(une_socket):
    """Apprend les pseudos presents et leur IP"""
    re_message = re.compile(rb'([^\:]+):([^\:]+):([^\:]*)\n')
    while(1):
        (data, (ip, port)) = une_socket.recvfrom(1024)
        if (data[0] == ':'):
            # on apprend un pseudo
            pseudo = data[1:]
            if (pseudo == mon_pseudo):
                continue
            if (not (pseudo in dico_pseudos)):
                print('nouveau pseudo %s a %s'%(pseudo, ip))
                dico_pseudos[pseudo] = ip
                sauver_liste_pseudos()
            continue
        resultat = re_message.search(data)
```

```

    if resultat :
        (expediteur, destinataire, message) = resultat.groups()
        print ('Reception de %s depuis %s'%(message, expediteur))
        if ((destinataire==mon_pseudo) or (destinataire=='*')):
            print ('%s > %s'%(expediteur, message))
            continue

def envoyer_message(pseudo, message, une_socket):
    print ("Envoi de %s a %s"%(message, pseudo))
    if (pseudo == ""):
        une_socket.sendto(b"%s:*:%s\n"%(mon_pseudo, message), TSAP)
        return
    if (not (pseudo in dico_pseudos)) :
        lire_liste_pseudos()
    if (not (pseudo in dico_pseudos)) :
        une_socket.sendto(b"%s:%s:%s\n"%(mon_pseudo, pseudo, message), TSAP)
        return
    print ("envoi cible de %s a %s"%(message, dico_pseudos[pseudo]))
    une_socket.sendto(b"%s:%s:%s\n"%(mon_pseudo, pseudo, message),
                      (dico_pseudos[pseudo], numero_port))

if os.fork():
    # je suis le pere
    re_message = re.compile(r'([^\:]+):([^\:]*)')
    while(1):
        saisie = input("destinataire:message>")
        resultat = re_message.search(saisie)
        if resultat:
            (destinataire, message) = resultat.groups()
            envoyer_message(destinataire, message.encode('utf8'), ma_socket_udp)
else :
    # je suis le premier fils
    if os.fork():
        # je suis toujours le premier fils
        apprendre_pseudos_et_afficher_message(ma_socket_udp)
    else:
        # je suis le second fils
        diffuser_pseudo(ma_socket_udp)

```