

TP 03

1 Introduction aux Sockets avec le protocole Daytime

1.1 Utilisation du service Daytime avec Telnet

- La commande `telnet time.nist.gov daytime` permet de se connecter à la machine time.nist.gov avec le protocole telnet et ensuite se connecter au protocole daytime. Cela a pour effet de d'afficher la date et l'heurs en temps univercel (UTC).
- Le protocole daytime est un protocole qui se trouve sur la couche application du modèl TCP/IP. Il peut fonctionner avec le protocole TCP ou UDP. Il utilise le port 13.
- Avec la commande `cat /etc/services | grep daytime` je trouve :

```
daytime      13/tcp
daytime      13/udp
```

Je testes donc de me connecter directement sur le port 13 avec la commande `netcat time.nist.gov 13` et j'obtiens le même résultat.

- Avec la commande `cat /etc/services | grep -w 'http\s\|ftp\s\|smtp\s\|telnet\s\|ssh\s\|echo\s'` je trouve les ports par défaut suivant :

```
echo          7/tcp
echo          7/udp
ftp           21/tcp
ssh           22/tcp          # SSH Remote Login Protocol
telnet        23/tcp
smtp          25/tcp          mail
http          80/tcp          www      # WorldWideWeb HTTP
at-echo       204/tcp          # AppleTalk echo
at-echo       204/udp
echo          4/ddp          # AppleTalk Echo Protocol
```

- On ne peut pas utiliser la commande `telnet` pour faire ces expériences car le protocole telnet existe seulement en mode connecter (en TCP). La principal différence entre le protocole TCP et le protocole UDP est que TCP demande un accusé de réception donc est plus fiable que UDP. En TCP, le service daytime établie une connexion pour donner la date et l'heur et s'assure que l'information est bien reçu. En UDP, le service daytime se contente juste d'envoyer la date et l'heur.

1.2 Programmation d'un client Daytime avec les Socket

- ✓
- ✓
- Réponse aux questions sur le programme

- La constantes AF_INET définit le type d'adresse utiliser donc le type de socket. Il existe AF_INET6 pour l'ipv6.
 - La constantes SOCK_STREAM définit le protocole de la couche transport utiliser. SOCK_STREAM pour TCP et SOCK_DGRAM pour UDP.
 - La variable s stock un objet de type socket. Cet objet permet d'effectuer la connexion sur le r.
 - La ligne 5 déclenchent la connexion.
 - La ligne 8 déclenchent la déconnexion TCP/IP.
 - La méthode recv (pour receive) prend en paramètre un nombre d'octets. Elle permet de lire la réponse de la taille du nombre d'octets.
 - Selon la documentation la méthode qui sert à envoyer des données est send ou sendall.
- ✓

2 Socket Python : requête HTTP à la main

2.1 Echauffons-nous d'abord avec Telnet

- ✓

2.2 Programmons notre client web !

- ✓
- ✓
- ✓
- ✓
- Lorsqu'on teste le programme avec le site web www.w3.org la réponse est tronqué car on récupère juste 1024 octets.

2.3 Raffinements

- ✓

3 Serveur Echo en Python

3.1 Petit rappel

- ✓

3.2 Version TCP

```
#!/usr/bin/python3.6
import os
import socket
import sys

try:
    host = ""
    port = 7777
```

```

s = socket.socket(socket.AF_INET6, socket.SOCK_STREAM, 0)
try :
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s.bind((host, port))
    s.listen(1)

    try :
        while True :
            # Récupère le socket de la connexion du client et sont ac
            sclient, a = s.accept()

            while True :
                data = sclient.recv(1500)
                if data == b"" : break
                sclient.send(data)

            print ("Requet from : ", a[0])

            sclient.close()

        except Exception as e :
            print("Error with accept : ", e)

    s.close()
except Exception as e :
    print("Error on bind the socket : ", e)

except KeyboardInterrupt:
    print('Interrupted')
    s.close()
    try:
        sys.exit(0)
    except SystemExit:
        os._exit(0)

```

- Pendant l'exécution, j'effectue la commande `netstat -tuap | grep 7777`:

```

tcp    0  0  localhost:46932  localhost:7777  ESTABLISHED 11059/nc
tcp6   0  0  [::]:7777       [::]:*          LISTEN       11058/python3.6
tcp6   0  0  localhost:7777  localhost:46932  ESTABLISHED 11058/python3.6

```

3.3 Version UDP (bonus)

- Le code :

```

#!/usr/bin/python3.6
import os
import socket
import sys

try:
    host = ""

```

```

port = 7777

s = socket.socket(socket.AF_INET6, socket.SOCK_DGRAM, 0)
try :
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s.bind((host, port))

    try :
        while True :
            data, a = s.recvfrom(1500)
            if data == b"" : break
            s.sendto(data, a)

            print ("Requet from : ", a[0])

    except Exception as e :
        print("Error with accept : ", e)

    s.close()
except Exception as e :
    print("Error on bind the socket : ", e)

except KeyboardInterrupt:
    print('Interrupted')
    s.close()
    try:
        sys.exit(0)
    except SystemExit:
        os._exit(0)

```

- Pendant l'exécution, j'effectue la commande `netstat -tuap | grep 7777:`

```

udp    0  0 localhost:40700  localhost:7777  ESTABLISHED 12713/nc
udp6   0  0 [::]:7777      [::]:*          12698/python3.6

```