



Trabajo Práctico Exploratorio Interface de Sockets

- 1) Descargue de los ejemplos de programas Python que utilizan la interface de Sockets.
- 2) En una instancia de terminal ejecute el servidor de echo sobre transporte UDP

```
# python udp_server.py
```

y en otra el cliente (no envíe nada):

```
# python udp_client.py
```
- 3) Verifique con el comando `netstat` procesos “escuchando” y conexiones establecidas. Intente identificar el socket de su programa (si hay). Revise los parámetros `t`, `l` y `n`.
- 4) Envíe un texto desde el cliente y explique el funcionamiento de ambos programas.
- 5) Repita el paso 2 pero realizando una captura con Wireshark. Filtre solo las PDU de su programa. ¿Observa lo esperado? Justifique.
- 6) Coordine con un compañero y ejecute el server en una máquina y el cliente en otra. Modifique los programas para reflejar estos cambios. Repita el paso 4. ¿Observa lo esperado? Justifique.
- 7) Repita los pasos 2, 3, 4 y 5 con el servidor y cliente TCP.
- 8) Pruebe el servidor `server_threads.py` con un cliente `telnet`. Describa el comportamiento. ¿Qué diferencias encuentra con los servidores de ejercicios anteriores?
- 9) Pruebe el servidor `server_select.py` con un cliente `telnet`. Describa el comportamiento. ¿Qué diferencias encuentra con los servidores de ejercicios anteriores?
- 10) Ejecute el servidor y cliente de chat. Verifique el mecanismo para “pasar” datos entre las conexiones. Modifique su programa para permitir un mensaje privado a un determinado usuario. ¿Cómo debe modificar su protocolo?
- 11) Existen dos programas mas de ejemplo: `pytraceroute.py` y `packet_sniffer.py`. Revise su arquitectura y verifique qué opciones de la interface de sockets se utilizan.
- 12) Ejecute el siguiente fragmento de programa para obtener opciones de la interface. Revise la documentación y explique qué significa cada una junto con un ejemplo de uso.

```
import socket
solist = [x for x in dir(socket) if x.startswith('SO_')]
solist.sort()
for x in solist:
    print x
```