

Cliente y servidor UDP en Python

Servidor

[crayon-6a9d4700b7f0d421328788/]

El servidor UDP es un programa que espera recibir paquetes de datos enviados por otros programas a través de una red. En este caso, se utiliza el módulo socket de Python para crear un socket UDP.

Para crear un servidor UDP, primero es necesario importar el módulo socket de Python. Luego, se crea un objeto socket para el protocolo de Internet (AF_INET) y el tipo de socket de datagrama (SOCK_DGRAM).

Una vez creado el objeto socket, se enlaza el socket al puerto y dirección IP donde el servidor estará escuchando. En este caso, se utiliza el puerto 5000 y la dirección IP localhost.

Después de enlazar el socket, se espera a recibir un mensaje entrante en el socket utilizando un bucle while. Cuando se recibe un mensaje, se imprime la cantidad de bytes recibidos y la dirección IP y puerto del cliente que envió el mensaje.

Finalmente, se envía una respuesta al cliente que envió el mensaje utilizando el método sendto del objeto socket. En este caso, se envía un mensaje de respuesta codificado en UTF-8.

Cliente

[crayon-6a9d4700b7f15084425616/]

El cliente UDP es un programa que envía paquetes de datos a un servidor a través de una red. En este caso, se utiliza el

módulo socket de Python para crear un socket UDP.

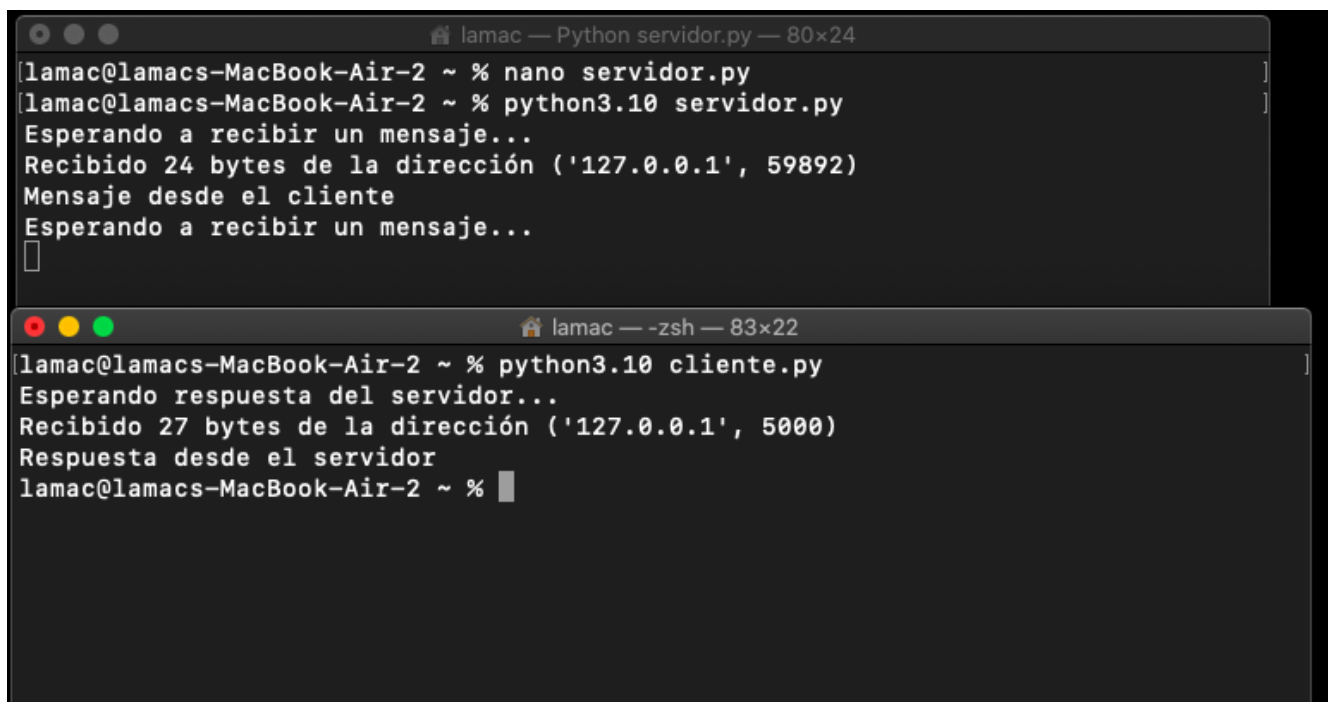
Para crear un cliente UDP, primero se importa el módulo socket de Python. Luego, se crea un objeto socket de tipo datagrama para el protocolo de Internet (AF_INET) y el tipo de socket de datagrama (SOCK_DGRAM).

Después de crear el objeto socket, se define la dirección IP y puerto del servidor al que se va a enviar el mensaje. En este caso, se utiliza la dirección IP localhost y el puerto 5000.

A continuación, se envía un mensaje al servidor utilizando el método sendto del objeto socket. En este caso, se envía un mensaje codificado en UTF-8.

Después de enviar el mensaje, el cliente espera una respuesta del servidor utilizando el método recvfrom del objeto socket. Cuando se recibe la respuesta, se imprime la cantidad de bytes recibidos y la dirección IP y puerto del servidor que envió la respuesta.

Finalmente, se cierra el socket del cliente.



```
lamac — Python servidor.py — 80x24
lamac@lamacs-MacBook-Air-2 ~ % nano servidor.py
lamac@lamacs-MacBook-Air-2 ~ % python3.10 servidor.py
Esperando a recibir un mensaje...
Recibido 24 bytes de la dirección ('127.0.0.1', 59892)
Mensaje desde el cliente
Esperando a recibir un mensaje...
█

lamac — zsh — 83x22
lamac@lamacs-MacBook-Air-2 ~ % python3.10 cliente.py
Esperando respuesta del servidor...
Recibido 27 bytes de la dirección ('127.0.0.1', 5000)
Respuesta desde el servidor
lamac@lamacs-MacBook-Air-2 ~ % █
```