

Actividad evaluada 1

Nombre: Patricio Espinoza A. **Sección:** 1

Ejemplo local

Se utilizo el archivo “txt_1.txt” de 500 KB, ejecutando el servidor y cliente:

Servidor \$ python3 server_echo_udp.py 1818

Cliente \$./client_echo3_udp.py 127.0.0.1 1818 < files/txt_1.txt > outputs/salida_1.txt

El archivo “salida_1.txt” resultó con un tamaño de 465KB al revisar sus propiedades. Sin embargo, un detalle es que en la previsualización del archivo aparecía con tamaño de 0 KB.

Cientes paralelos

Usando el mismo servidor, dos archivos “txt_1.txt” y “txt_2.txt” con tamaños de 500 KB se enviaron en simultaneo, obteniendo los archivos “salida_2.txt” y “salida_3.txt”, al ver su tamaño en propiedades pasaron a 298KB y 290KB respectivamente. Los archivos aparecían con tamaños de 0 KB inicialmente en la previsualización.

Cientes

\$./client_echo3_udp.py 127.0.0.1 1818 < files/txt_1.txt > outputs/salida_2.txt &

./client_echo3_udp.py 127.0.0.1 1818 < files/txt_2.txt > outputs/salida_3.txt &

wait

Cliente paralelos (Binarios)

Cientes

\$./client_echo3_udp_binario.py 127.0.0.1 1818 < files/bin_1.bin >
outputs/salida_1.bin &

./client_echo3_udp_binario.py 127.0.0.1 1818 < files/bin_2.bin > outputs/salida_2.bin
&

Wait

Usando el mismo servidor, dos archivos binarios “bin_1.bin” y “bin_2.bin” con tamaños de 500 KB se enviaron en simultaneo, obteniendo “salida_1.bin” con un tamaño de 44.9KB y “salida_2.bin” con un tamaño de 38.5KB en sus propiedades. Los archivos aparecían con tamaños de 0 KB inicialmente en la previsualización.

Pérdida de datos

Por una parte, se establece un límite de 1500 bytes, y por tanto mensajes más grandes no se recibirán correctamente.

- `s.recv(1500)` # Modificar esto

No hay un mecanismo para una sincronización correcta en la que se asegure que distintos clientes (hilos) no accedan a recursos compartidos simultáneamente (consola o socket).

- Se deberían usar los Locks

El `time.sleep(3)` depende de si durante cierto tiempo no se envían datos al servidor, lo cual puede ser incorrecto en un contexto donde, no todos los datos han sido enviados debido a una latencia y por tanto ocurre que pasa el tiempo establecido, se cierra la conexión y el cliente ya no pudo enviar más información.