

Control 1

Redes

Plazo de entrega: 6 de septiembre de 2023

José M. Piquer

Intro

El control se responde en forma individual, sin preguntas entre Uds o a terceros. Todas las dudas pregúntenlas en el foro de U-cursos y el equipo docente les responderá. Entreguen las tres preguntas en un PDF con su nombre por U-cursos. Pueden (y deben) buscar material en Internet para responder, pero citen siempre las fuentes que utilizan. Pueden usar chatGPT para ayudarlos, pero, no pueden entregar su respuesta directamente, deben explicar y analizar todo lo que les propone. Recuerden que chatGPT se equivoca, alucina y miente con mucha convicción. Siempre recuerden mencionarlo si lo utilizan.

P1: sockets

Suponga que existen muchos servidores en Internet que implementan un servicio de cálculo de un número entero que demora mucho tiempo. Algunos servidores son más rápidos, pero no siempre precisos, y otros son más lentos pero con mejores resultados. Su misión es escribir un cliente que se conecte a una lista de servidores y, a medida que vaya recibiendo respuestas, las vaya promediando y mostrando el resultado hasta este momento. Cuando haya recibido todas las respuestas, termina.

Ojo que no sabemos qué servidor va a responder antes, por lo que debemos ir siempre leyendo del que está listo para enviar su resultado. Está prohibido usar '*busy waiting*', es decir deben bloquearse cuando ningún socket tiene datos.

Para simplificar el protocolo, supongamos que los servidores simplemente esperan la conexión y cuando la reciben (sin ningún dato) calculan el número y envían la respuesta codificada como un string UTF-8, y cierran la conexión.

Un ejemplo de servidor de prueba (que siempre retorna 20, pero demora un tiempo aleatorio entre 0 y 3 segundos):

```
#!/usr/bin/python3
import jsockets
import sys
import time, random

def calc():
    # demoro un tiempo entre 0 y 3 s
    time.sleep(random.random()*3.0)
    return 20

# main
if len(sys.argv) != 2:
    print('Use: '+sys.argv[0]+' port')
    sys.exit(1)

s = jsockets.socket_tcp_bind(int(sys.argv[1]))
if s is None:
    print('could not open socket')
    sys.exit(1)

while True:
    conn, addr = s.accept();
    print('Connected by', addr)
    conn.send(str(calc()).encode('UTF-8'))
    print(addr, 'result sent')
    conn.close()
```

Entregue el código de su cliente. Puede usar el servidor escrito acá para probar su cliente si quiere asegurarse que funcione. Vídeo de ejemplo de uso: <https://youtu.be/atZRYzEHIRg>

P2: Proxies

En la actividad del curso del proxy-copy, vimos que funciona bien con http, pero no siempre funciona con https. Responda las siguientes preguntas (el problema es complejo y hay mucho material en Internet. No se les pide una tesis acá, solo que expliquen los temas técnicos de base y un bosquejo de las soluciones):

1. ¿Funciona con SMTP (el protocolo de mail)?
2. ¿Por qué no funciona con HTTPS?
3. Busque distintas formas de hacer un proxy https y discuta cuál sería la mejor
4. Si tengo un proxy para https, ¿puedo guardar un log con todo el tráfico generado en esa sesión, como hicimos en la actividad?
5. Si tengo el log, ¿podré verlo descriptado?

P3: Servidores

Una empresa de video juegos está implementando su juego en un servidor que debe soportar múltiples jugadores compartiendo un escenario, donde los jugadores deben poder verse, interactuar con el entorno, y todos ver una versión coherente del mundo virtual.

Tienen cuatro propuestas de implementación que les han traído sus ingenieros:

1. Usar procesos pesados: cada cliente es atendido por un servidor dedicado, con un socket TCP (SOCK_STREAM) para cada uno.
2. Usar threads: cada cliente es atendido por un thread dedicado, con un socket TCP (SOCK_STREAM) para cada uno.
3. Usar select: todos los clientes son atendidos por el mismo proceso que usar select() para definir a qué socket atender, cada cliente tiene un socket TCP (SOCK_STREAM) propio.
4. Socket común: todos los clientes comparten el mismo socket UDP (SOCK_DGRAM) para enviar sus requerimientos, el servidor es un proceso que va atendiendo los requerimientos uno por uno.

Analice las distintas alternativas y discuta sus pro y contras. Elija la alternativa que Ud usaría en un caso así y justifique su elección.