África Franco Montero
Isabel Gude Gómez

2º GREI
Grupo 3

Informe Entrega 2

Apartado 1 c:

En primer lugar añadimos en el cliente un sleep(5) para que este espere 5 segundos antes de recibir el mensaje y de esa forma le dé tiempo al servidor a enviar los dos mensajes. Este cambio se realiza justo antes de la función recv como se ve a continuación:

```
sleep(5);

//Recibimos el mensaje del servidor
if ( (nbytes=recv(socketcli, mensaje, 1000,0)) < 0) {
    perror ( "No se pudo recibir el mensaje" ) ;
    exit ( EXIT_FAILURE ) ;
}
else if (nbytes==0){
    perror("Se ha cerrado el socket de conexión");
    exit ( EXIT_FAILURE ) ;
}
mensaje[nbytes]='\0';
printf("Se ha recibido -%s- con %zd bytes\n", mensaje, nbytes);
```

Tras esto, en el servidor añadimos otra función send, con esta enviamos un segundo mensaje. La nueva función se introduce justo debajo la primera como podemos ver en la siguiente imagen:

```
//Enviamos varios mensajes a el cliente
if ( (nbytes1=send(socketcon, mensaje, strlen(mensaje),0)) < 0) {
    perror ( "No se pudo enviar el mensaje" ) ;
    exit ( EXIT_FAILURE ) ;
}

if ( (nbytes2=send(socketcon, mensaje2, strlen(mensaje),0)) < 0) {
    perror ( "No se pudo enviar el mensaje" ) ;
    exit ( EXIT_FAILURE ) ;
}
printf ("Número de bytes enviados %zd\n", nbytes1+nbytes2);
```

Finalmente, esto es lo que podemos ver al ejecutar los programas:

-Servidor:

```
africa@PCAFM:~/Redes/Practica2.1$ ./serv 7006
Dirección IP del cliente: 37.127.0.0
Puerto del cliente: 13298
Número de bytes enviados 26
```

-Cliente:

```
africa@PCAFM:~/Redes/Practica2.1$ ./cli 127.0.0.1 7000
Se ha recibido -Hola caracolaAdios caracol- con 26 bytes
```

Se puede observar que en el cliente se envían los dos mensajes consecutivos (“Hola caracola” y “Adios caracol”) y el número de bytes enviados y recibidos es la suma de los bytes de cada mensaje.

Si eliminamos la función sleep del cliente, pese a que el servidor siempre envía los dos mensajes, no podemos asegurar que ambos se envíen antes de que el cliente los reciba. Por esto hay veces en las que solo se recibe el primer mensaje, y veces en las que se reciben los dos.

En la siguiente imagen se puede ver como ejecutando dos veces el mismo código, la primera vez se reciben los dos mensajes y la segunda solo uno.

```
africa@PCAFM:~/Redes/Practica2.1$ ./cli 127.0.0.1 7001
Se ha recibido -Hola caracolaAdios caracol- con 26 bytes
africa@PCAFM:~/Redes/Practica2.1$ ./cli 127.0.0.1 7002
Se ha recibido -Hola caracola- con 13 bytes
```

Mientras que el servidor siempre envía los dos mensajes:

```
africa@PCAFM:~/Redes/Practica2.1$ ./serv 7001
Dirección IP del cliente: 127.0.0.1
Puerto del cliente: 63663
Número de bytes enviados 26
africa@PCAFM:~/Redes/Practica2.1$ ./serv 7002
Dirección IP del cliente: 127.0.0.1
Puerto del cliente: 16616
Número de bytes enviados 26
```

Apartado 1 d:

Tras añadir el siguiente código que se adjunta a continuación:

```
//Recibimos el mensaje del servidor
while( (nbytes=recv(socketcli,mensaje,5,0)) > 0 ){
    mensaje[nbytes]='\0';
    printf("Se ha recibido -%s- con %zd bytes\n", mensaje, nbytes);

}
```

Podemos comprobar que el mensaje o mensajes enviados por el servidor son recibidos en grupos de x bytes siendo x el número que se especifica en la función recv como número de bytes a recibir. Este bucle termina cuando se acaba de recibir el último mensaje. A continuación unos ejemplos:

```
//Recibimos el mensaje del servidor
while( (nbytes=recv(socketcli,mensaje,5,0)) > 0 ){
    mensaje[nbytes]='\0';
    printf("Se ha recibido -%s- con %zd bytes\n", mensaje, nbytes);

}
```

```
africa@PCAFM:~/Redes/Practica2.1$ ./cli 127.0.0.1 7000
Se ha recibido -Hola - con 5 bytes
Se ha recibido -carac- con 5 bytes
Se ha recibido -olaAd- con 5 bytes
Se ha recibido -ios c- con 5 bytes
Se ha recibido -araco- con 5 bytes
Se ha recibido -l- con 1 bytes
```

```
while( (nbytes=recv(socketcli,mensaje,10,0)) > 0 ){
    mensaje[nbytes]='\0';
    printf("Se ha recibido -%s- con %zd bytes\n", mensaje, nbytes);

}
```

```
africa@PCAFM:~/Redes/Practica2.1$ ./cli 127.0.0.1 7001
Se ha recibido -Hola carac- con 10 bytes
Se ha recibido -olaAdios c- con 10 bytes
Se ha recibido -aracol- con 6 bytes
```

África Franco Montero
Isabel Gude Gómez

2º GREI
Grupo 3

Ejercicio 3:

Si introducimos un sleep en el lazo del cliente donde se van leyendo las líneas del archivo nos da tiempo de lanzar un segundo cliente en otra terminal. De esta forma el servidor es capaz de atender a los dos clientes a través de la misma conexión por el mismo puerto.

En primer lugar lanzamos el primer cliente:

```
africa@PCAFM:~/Redes/Practica2.2$ ./cli texto.txt 127.0.0.1 7000
Se ha recibido PRIMER CLIENTE 1
Se ha recibido PRIMER CLIENTE 2
Se ha recibido PRIMER CLIENTE 3
```

Seguidamente lanzamos el segundo:

```
africa@PCAFM:~/Redes/Practica2.2$ ./cli texto2.txt 127.0.0.1 7000
Se ha recibido SEGUNDO CLIENTE 1
Se ha recibido SEGUNDO CLIENTE 2
Se ha recibido SEGUNDO CLIENTE 3
```

El servidor atiende al primer cliente y después al segundo:

```
africa@PCAFM:~/Redes/Practica2.2$ ./serv 7000
Dirección IP del cliente: 127.0.0.1
Puerto del cliente: 45248
Se ha recibido primer cliente 1
Se ha recibido primer cliente 2
Se ha recibido primer cliente 3
Dirección IP del cliente: 127.0.0.1
Puerto del cliente: 9440
Se ha recibido segundo cliente 1
Se ha recibido segundo cliente 2
Se ha recibido segundo cliente 3
```