Sistemas Distribuidos

Grado de Ingeniería en Informática. Curso 2022-2023

Ejercicio evaluable 1 - Sockets

Grupo reducido: **82**
Letra del equipo: **J**

Práctica realizada por el alumno:

Binxian Huang: 100451010@alumnos.uc3m.es

Diseño y estructura del proyecto:

- servidor.c:

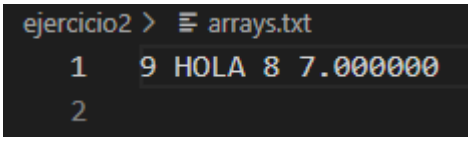
Para la parte del servidor, se crea el socket con opción de reutilizar la dirección IP usada, para evitar un posible error al reintentar usar la misma dirección. El puerto proporcionado por el usuario se guarda en una estructura de tipo *sockaddr_in* del servidor en formato de red *Big Endian* y dirección IP cualquiera disponible. Después se enlaza al socket creado y se escucha por el puerto esperando conexiones del cliente. Cuando se recibe una conexión y se conecta correctamente con el cliente, se crea un hilo que ejecuta la petición del cliente mientras que el servidor continúa esperando peticiones de conexión. En cuanto al envío y recepción de mensajes, se realizan mediante cadenas de caracteres, con las funciones *sendMessage* y *readLine* enviando y recibiendo cada valor por separado. A la función *sendMessage* se proporciona el descriptor de socket creado al aceptar la conexión, el buffer de texto donde está el mensaje a enviar y el tamaño a enviar. La función envía datos iterativamente hasta que no quede ninguno. La función *readLine* funciona de manera parecida, se proporciona el mismo descriptor de socket, el buffer donde se guardará el mensaje leído y el tamaño del mensaje que se desea leer. Esta función también lee recursivamente byte a byte hasta que no quede ningún byte por leer. En el envío, si el dato es un entero o flotante se escribe en el buffer de texto en forma de array mientras que si es una cadena de caracteres, se proporciona directamente la el puntero a la cadena, y en la lectura si es un número se usa la función *atoi* o *atof*, y en caso de ser una cadena se proporciona también la dirección de memoria de la variable donde se guardará.

- claves.c:

Para la parte del cliente, se proporciona el nombre del servidor y el puerto donde escuchará mediante variables de entorno, que serán leídas y guardadas en variables locales. Mediante la función gethostbyname se obtienen los datos del servidor, entre los que está la dirección IP necesaria para conectar el socket creado del cliente con el servidor. Tras establecer la conexión, el cliente realizará la petición, esperará la respuesta, y tras recibir la respuesta cerrará la conexión cerrando su socket. La conexión y cierre se realizará para cada petición que realice el cliente. En cuanto al envío y recepción de mensajes, se realiza de la misma forma que el servidor, con las funciones *sendMessage* y *readLine*, funcionando de la misma manera.

Pruebas:

En cuanto a las pruebas, no se ha realizado ningún cambio por lo que el funcionamiento es el mismo:



```
juliohuang@JULIO-H:/mnt/d/Julio/Uc3m/Curso 3/Cuatrimestre 2/SistemasDistribuidos/labs_
SSDD/ejercicio2$ ./cliente
These are the operations you can use:
init - 0
set_value - 1 <key> <value1> <value2> <value3>
get_value - 2 <key>
modify_value - 3 <key> <value1> <value2> <value3>
delete_key - 4 <key>
exist - 5 <key>
copy_key - 6 <key1> <key2>
EXIT - 7
Operation code: 0
Client socket created.
Connection established.
Operation code 0 sent correctly.
Init response result received correctly with value 1.
Connection socket closed of client init.
Initilization correct.
Operation code: 1
Arguments for set_value: 9 "HOLA" 8 7
Client socket created.
Connection established.
Operation code 1 sent correctly.
Key 9 sent correctly.
Value1 HOLA sent correctly.
Value2 8 sent correctly.
Value3 7.000000 sent correctly.
Set_value response result received correctly with value 1.
Conection socket closed of client set_value.
The value with key 9 set correctly.
Operation code:
```

```
SDD/ejercicio2$ ./servidor
Server socket created.
Server socket binded.
Listening on port 8000.
Connection accepted.
Operation code received on server: 0
Init response result sent correctly with value: 1
Connection socket closed in server.
Connection accepted.
Operation code received on server: 1
Key of set_value petition received on server with value: 9
Value1 of set_value petition received on server with value: HOLA
Value2 of set_value petition received on server with value: 8
Value3 of set_value petition received on server with value: 7.000000
Key to set: 9, Value1: HOLA, Value2: 8, Value3: 7.000000
Value set correctly: 9 HOLA 8 7.000000
Set_value response result sent correctly with value: 1
Connection socket closed in server.
```

```
Operation code: 1
Arguments for set_value: 9 "ADIOS" 8 7
Client socket created.
Connection established.
Operation code 1 sent correctly.
Key 9 sent correctly.
Value1 ADIOS sent correctly.
Value2 8 sent correctly.
Value3 7.000000 sent correctly.
Set_value response result received correctly with value 0.
Conection socket closed of client set_value.
An error ocurred while setting the value with key 9.
Operation code:
```

```
Connection accepted.
Operation code received on server: 1
Key of set_value petition received on server with value: 9
Value1 of set_value petition received on server with value: ADIOS
Value2 of set_value petition received on server with value: 8
Value3 of set_value petition received on server with value: 7.000000
Key to set: 9, Value1: ADIOS, Value2: 8, Value3: 7.000000
Key already exists on set_value.: Success
Set_value response result sent correctly with value: 0
Connection socket closed in server.
```

Con *init(0)* se inicializa el fichero y con *set_value(1)* se asigna los valores a la key correspondiente, y en caso de que la key ya tenga valores asignados salta error.

```
Operation code: 2
Arguments for get_value: 9
Client socket created.
Connection established.
Operation code 2 sent correctly.
Key 9 sent correctly.
Get_value response result received correctly with value 1.
Value1 received correctly with value HOLA.
Value2 received correctly with value 8.
Value3 received correctly with value 7.000000.
Conection socket closed in client get_value.
The value with key 9 is: HOLA 8 7.000000
Operation code: 2
Arguments for get_value: 1
Client socket created.
Connection established.
Operation code 2 sent correctly.
Key 1 sent correctly.
Get_value response result received correctly with value 0.
Conection socket closed in client get_value.
An error ocurred while getting the value with key 1.
```

```
Connection accepted.
Operation code received on server: 2
Key of get_value petition received on server with value: 9
Key to get: 9
Get_value response result sent correctly with value: 1
Get_value response value1 sent correctly with value: HOLA
Get_value response value2 sent correctly with value: 8
Get_value response value3 sent correctly with value: 7.000000
Connection socket closed in server.
Connection accepted.
Operation code received on server: 2
Key of get_value petition received on server with value: 1
Key to get: 1
Key does not exist on get_value.: Success
Get_value response result sent correctly with value: 0
Connection socket closed in server.
```

Con *get_value(2)* se obtienen los valores guardados correspondientes a la key que se pide, y devuelve error en caso de que no esté guardada la key que se pide.

```
Operation code: 3
Arguments for modify_value: 9 "ADIOS" 9 9
Client socket created.
Connection established.
Operation code 3 sent correctly.
Key 9 sent correctly.
Value1 ADIOS sent correctly.
Value2 9 sent correctly.
Value3 9.000000 sent correctly.
Modify_value response result received correctly with value 1.
Conection socket closed in client modify_value.
The value with key 9 modified correctly.
Operation code: You must introduce one operation code.
Operation code: 3
Arguments for modify_value: 1 "HOLA" 1 1
Client socket created.
Connection established.
Operation code 3 sent correctly.
Key 1 sent correctly.
Value1 HOLA sent correctly.
Value2 1 sent correctly.
Value3 1.000000 sent correctly.
Modify_value response result received correctly with value 0.
Conection socket closed in client modify_value.
An error ocurred while modifying the value with key 1.
```

```
Connection accepted.
Operation code received on server: 3
Key of modify_value petition received on server with value: 9
Value1 of modify_value petition received on server with value: ADIOS
Value2 of modify_value petition received on server with value: 9
Value3 of modify_value petition received on server with value: 9.000000
Key to modify: 9, Value1: ADIOS, Value2: 9, Value3: 9.000000
Modify_value response result sent correctly with value: 1
Connection socket closed in server.
Connection accepted.
Operation code received on server: 3
Key of modify_value petition received on server with value: 1
Value1 of modify_value petition received on server with value: HOLA
Value2 of modify_value petition received on server with value: 1
Value3 of modify_value petition received on server with value: 1.000000
Key to modify: 1, Value1: HOLA, Value2: 1, Value3: 1.000000
Key does not exist on modify_value.: Success
Modify_value response result sent correctly with value: 0
Connection socket closed in server.
```

ejercicio2 > arrays.txt

```
1   9 ADIOS 9 9.000000
2
```

Con *modify_value(3)* se modifica los valores de una key ya existente con unos valores nuevos, devolviendo también error en caso de que la key que se desea modificar tampoco esté guardada.

```
Operation code: 5
Arguments for exist: 9
Client socket created.
Connection established.
Operation code 5 sent correctly.
Key 9 sent correctly.
Exist response result received correctly with value 1.
Conection socket closed in client exist.
The value with key 9 exists.
Operation code: 5
Arguments for exist: 1
Client socket created.
Connection established.
Operation code 5 sent correctly.
Key 1 sent correctly.
Exist response result received correctly with value 0.
Conection socket closed in client exist.
The value with key 1 does not exist.
Operation code:
```

```
Connection accepted.
Operation code received on server: 5
Key of exist petition received on server with value: 9
Key to verify if exist: 9
Exist response result sent correctly with value: 1
Connection socket closed in server.
Connection accepted.
Operation code received on server: 5
Key of exist petition received on server with value: 1
Key to verify if exist: 1
Exist response result sent correctly with value: 0
Connection socket closed in server.
```

Con *exist(5)* se comprueba que la key que se proporciona existe o no, en caso de que exista o no se devuelve el mensaje correspondiente, y error en otro caso.

```
Operation code: 6
Arguments for copy_key: 9 2
Client socket created.
Connection established.
Operation code 6 sent correctly.
Key1 9 sent correctly.
Key2 2 sent correctly.
Copy_key response result received correctly with value 1.
Conection socket closed in client copy_key.
The value with key 9 copied correctly to key 2.
Operation code: 6
Arguments for copy_key: 1 9
Client socket created.
Connection established.
Operation code 6 sent correctly.
Key1 1 sent correctly.
Key2 9 sent correctly.
Copy_key response result received correctly with value 1.
Conection socket closed in client copy_key.
The value with key 1 copied correctly to key 9.
```

```
Connection accepted.
Operation code received on server: 6
Key1 of copy_key petition received on server with value: 9
Key2 of copy_key petition received on server with value: 2
Key to copy for: 9. Key to copy at: 2.
Value set correctly: 2 ADIOS 9 9.000000
Copy_key response result sent correctly with value: 1
Connection socket closed in server.
Connection accepted.
Operation code received on server: 6
Key1 of copy_key petition received on server with value: 1
Key2 of copy_key petition received on server with value: 9
Key to copy for: 1. Key to copy at: 9.
Copy_key response result sent correctly with value: 1
Connection socket closed in server.
```

ejercicio2 > arrays.txt

```
1   9 HOLA 1 1.000000
2   1 HOLA 1 1.000000
3   2 ADIOS 9 9.000000
4
```

Con *copy_key(6)* se copia los valores de la primera key en la segunda en caso de que ambas existan, se crea la segunda key en caso de que no exista, y en caso de que la primera no exista se devuelve error.

```
Operation code: 4
Arguments for delete_key: 9
Client socket created.
Connection established.
Operation code 4 sent correctly.
Key 9 sent correctly.
Delete_key response result received correctly with value 1.
Conection socket closed in client delete_key.
The value with key 9 deleted correctly.
Operation code: 4
Arguments for delete_key: 3
Client socket created.
Connection established.
Operation code 4 sent correctly.
Key 3 sent correctly.
Delete_key response result received correctly with value 0.
Conection socket closed in client delete_key.
An error ocurred while deleting the value with key 3.
```

```
Connection accepted.
Operation code received on server: 4
Key of delete_key petition received on server with value: 9
Key to delete: 9
Delete_key response result sent correctly with value: 1
Connection socket closed in server.
Connection accepted.
Operation code received on server: 4
Key of delete_key petition received on server with value: 3
Key to delete: 3
Key does not exist on delete_key.: Success
Delete_key response result sent correctly with value: 0
Connection socket closed in server.
```

ejercicio2 > arrays.txt

```
1   9 ADIOS 9 9.000000
2   1 HOLA 1 1.000000
3
```

Con *delete_key(4)* se elimina la key proporcionada y sus valores correspondientes, y devuelve error en caso de que la key a eliminar no esté guardada.

Diseño y estructura del proyecto:

- compilar:

make

- ejecutar servidor:

./servidor 8080

- ejecutar cliente:

env LD_LIBRARY_PATH=$LD_LIBRARY_PATH:. IP_TUPLAS=localhost PORT_TUPLAS=8080 ./cliente