

Examen SO.

18 de enero de 2022, de 15:00 a 17:00

Programación en C

Se desea implementar un servidor de subscripciones. Los clientes que se conectan pueden subscribirse a un tema concreto (una palabra) y también hacer una publicación asociada a un tema, de manera que todos los clientes suscritos a ese tema recibirán esa publicación. Para implementar este servidor necesitamos una lista de subscripciones.

1. Escribe la declaración de una estructura de datos que permita almacenar los datos de 100 subscripciones (una lista de subscripciones). Para cada subscripción queremos tener el socket del cliente que se subscribe (un entero) y el tema al que el cliente se subscribe (una palabra),
2. Escribe el código de una función que reciba como parámetro la lista de subscripciones y un tema concreto (una palabra) y proporcione un vector con los sockets de todos los clientes que estén suscritos a ese tema. Además, retornará el número de clientes suscritos al tema.
3. Escribe el código de un programa principal que declare una lista de subscripciones. A continuación, suponiendo que ya hay datos en la lista, el programa pedirá al usuario que introduzca por teclado un tema (una palabra) y obtendrá, usando la función del apartado 2, los sockets de los clientes suscritos a ese tema. Finalmente, escribirá en consola los sockets de los clientes suscritos (un socket en cada línea de consola).

Bases de Datos Multinivel

Necesitamos una base de datos sobre usuarios, temas y subscripciones. Para cada usuario queremos tener un identificador, su nombre (una palabra) y su email. Para cada tema queremos tener su identificador, una palabra clave y una descripción. Los usuarios pueden subscribirse a temas de su interés. Cada usuario puede estar suscrito a varios temas y cada tema puede tener varios usuarios suscritos.

1. Escribir las sentencias SQL necesarias para crear las tablas correspondientes a la base de datos que se desea implementar.
2. Escribir la sentencia SQL necesaria para realizar la siguiente consulta:
“Dame la descripción de los temas a los que están suscritos tanto ‘Juan’ como ‘Pedro’ (tienen que estar suscritos ambos)”.
3. Escribir el código en C necesario para pedir por teclado una palabra clave correspondiente a un tema y escribir en pantalla los emails de los usuarios suscritos a ese tema (un email por línea de consola).

Linux

1. Escribe el comando (o secuencia de comandos) necesarios hacer una copia del fichero datos.txt con el nombre datos.bak y quitarle a esa copia los permisos de escritura.
2. Escribe el código de un script que tenga como parámetro una palabra. El script debe escribir en consola el número de ficheros que hay en el directorio de trabajo que tienen más de 3 líneas en las que aparece la palabra que figura como parámetro del script.

Servidor

Se desea implementar un servidor de publicaciones-subscripciones (servidor pub-sub). Cualquier cliente que se conecte al servidor puede subscribirse a un tema (una palabra) o hacer una publicación (enviar una frase) asociada a un tema concreto. Esa publicación será enviada por el servidor a todos los clientes suscritos al tema. Los clientes pueden subscribirse a varios temas, dar de baja su subscripción a cualquiera de los temas a los que están suscritos o desconectarse.

En resumen, cualquier cliente que se conecte al servidor puede hacer cualquiera de las operaciones siguientes:

- a) Subscribirse a un tema. Para ello enviará al servidor el tema (una palabra). El servidor no enviará ninguna respuesta.
- b) Hacer una publicación. Para ello enviará al servidor una publicación (una frase) y un tema. El servidor enviará la publicación a todos los clientes suscritos a ese tema. En el caso de que no haya ningún cliente suscrito al tema el servidor enviará como respuesta una indicación informando de tal circunstancia.
- c) Dar de baja una subscripción. El cliente enviará el tema del cual quiere darse de baja. En el caso de que el cliente no esté suscrito a ese tema el servidor responderá con una indicación informando de tal circunstancia.
- d) Desconectarse. El servidor dará de baja todas las subscripciones de ese cliente.

Para implementar el servidor pueden usarse las funciones que se describen a continuación, que ya están implementadas. Las funciones reciben como parámetro una estructura de tipo `ListaSubscripciones` (ya definida), compartida por todos los threads, en la que se guarda información de todas las subscripciones (socket de cada cliente y tema al que se ha suscrito).

```
void PonSubscripcion (ListaSubscripciones *lista, char tema[20], int socket)
/* Añade a la lista una nueva subscripción (socket del cliente que se subscribe + tema) */

int QuitaSubscripcion (ListaSubscripciones *lista, char tema[20], int socket)
/* Elimina de la lista la subscripción. Retorna 0 si ok y -1 si el cliente identificado con el socket no
está suscrito al tema*/

int DameSuscritos (ListaSubscripciones *lista, char tema[20], int socket[100])
/* Coloca en el vector de sockets los sockets de todos los clientes suscritos al tema y retorna el número
de clientes suscritos */

void QuitaTodas (ListaSubscripciones *lista, int socket)
/* Elimina de la lista todas las subscripciones del cliente identificado con el socket*/
```

Se pide lo siguiente:

1. Especificar un posible protocolo de aplicación para el servidor descrito.
2. Indicar qué variables o estructuras de datos globales se utilizan para implementar el servidor.
3. Escribir el código de la función `*AtenderCliente` que debe ejecutar cada uno de los threads que se pongan en marcha en el servidor. Incluir las operaciones necesarias para garantizar exclusión mutua cuando sea necesario.

Cliente

Escribir el código de un cliente que pueda entenderse con el servidor descrito en el ejercicio anterior, y en coherencia con la solución que se ha dado a ese ejercicio anterior (es decir, respetando el protocolo de aplicación que se ha establecido).

En concreto, se pide:

- Dibujar el formulario del cliente indicando claramente el nombre de sus elementos clave (botones, cuadros de texto, labels, etc.). Tener en cuenta que el cliente NO PUEDE usar `MessageBox`.
- Escribir el código de todas las funciones del cliente.
- Incorporar los mecanismos necesarios para evitar que se produzca la excepción cross-threading.