

PRACTICA FINAL

Diseño de un servidor Chat multiusuario

Descripción de la práctica

Realice un programa que permita conectar a un número determinado de clientes, usando el cliente de la práctica anterior, para charlar entre ellos de forma que todos lean lo que escriben los demás. Las especificaciones serán las siguientes.

FORMATO

```
chatserver [-p <puerto>] [-n <numero de clientes>]
```

DESCRIPCION

El programa debe quedarse indefinidamente escuchando en el puerto especificado y permitirá mantener hasta **n** conexiones simultáneas de clientes.

Cada cliente puede enviar mensajes que son retransmitidos a todos los demás clientes que estén conectados.

Todos los usuarios pueden escuchar los mensajes de los demás pero sólo los que se han asignado un nombre pueden hablar. Cada cliente puede asignarse un nombre con el comando `\nombre` y una vez que lo haya hecho, aparecerá en las listas de usuarios y podrá enviar mensajes y recibir mensajes privados.

El servidor, al retransmitir los mensajes de un usuario al resto de usuarios, añadirá el nombre del usuario que ha escrito dicho mensaje, para que se sepa quién está hablando.

Además se considerará que si un cliente envía una línea que empieza con el carácter `\`, la línea no es una frase para retransmitir, sino que es un comando que debe ser interpretado por el servidor. Puede añadir los comandos que considere útiles, pero estos son los que deben estar:

```
\nombre <nick>
```

El usuario elige su nombre o lo cambia

```
\lista
```

El servidor envía a ese cliente una lista de todos los usuarios conectados que se han asignado nombres y además dice cuántos usuarios invisibles (que no se han asignado nombre todavía, pero están escuchando) hay en el sistema.

```
\m <usuario> Mensaje...
```

Envía el mensaje sólo al usuario de nombre especificado. En el caso de que un cliente cierre la conexión, por ejemplo haciendo `^C`, el servidor debe darse cuenta y reutilizar su sitio para atender a otros posibles clientes. Si un cliente intenta conectarse cuando todos los sitios están ocupados se le debe notificar con un mensaje.

EJEMPLOS DE USO

Para lanzar el servidor en el puerto 6789 permitiendo hablar a 4 usuarios:

En 10.1.1.26] % chatserver -p 6789 -n 4

Mientras el servidor chatserver está funcionando se puede usar con el programa chat de la práctica anterior. Por ejemplo veamos varias conexiones (con > marcamos lo que escribe el usuario)

En 10.1.1.27	En 10.1.1.28
% chat -c 10.1.1.26 6789	% chat -c 10.1.1.26 6789
Conectado al servidor de chat	Conectado al servidor de Chat
>¿Hay alguien ahí?	
Elige un nombre antes de hablar	
>\nombre A	
>¿Hay alguien ahí?	
A: ¿Hay alguien ahí?	A: ¿Hay alguien ahí?
	>Hola
	Elige un nombre antes de hablar
>\lista	
Usuarios conectados:	
A y 1 invisible.	
	>\nombre B
Nuevo usuario visible B	
B: Hola	>Hola
>\lista	B: Hola
Usuarios conectados:	
A, B y 0 invisibles	

Consideraciones adicionales obligatorias

Para la gestión y encuesta de sockets el alumno deberá utilizar la función **poll()** en vez de la función **select()**.