

PRÁCTICA 1: PROGRAMACIÓN CON SOCKETS

Héctor Menéndez

AIDA Research Group
Computer Science Department
Universidad Autónoma de Madrid

30 de enero de 2013

Index

- 1 Objetivos
- 2 Ejercicio 1: Chat de texto IRC
- 3 Ejercicio 2: Chat de audio UDP

Objetivos

- 1 Familiarizarse con los mecanismos de comunicación mediante sockets.
- 2 Comprender de forma práctica las ventajas y desventajas de implementar un sistema de comunicaciones sobre distintos protocolos de transporte.
- 3 Seguir la descripción de un protocolo de aplicación para comunicarse con un sistema remoto.
- 4 **IRC no es un objetivo.**

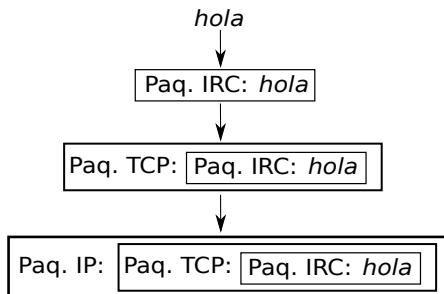
Recomendación

- Para el correcto funcionamiento de la practicas es recomendable tener habilidad en la programacion con hilos o procesos

Ejercicio 1: Chat de texto IRC

- Se utilizará el protocolo IRC.
- Como en toda conexión TCP/IP necesitamos la IP y el puerto del servidor.
- Hay que implementar un cliente:
 - En modo texto.
 - Con una interfaz GTK.
- Se puede elegir una de las dos opciones.
- Se programará en C.

Ejercicio 1: Encapsulamiento IRC



Ejercicio 1: Ejecución

clienteIRC <servidor> <puerto> <usuario> <canal>

- **servidor**: es el servidor al que conectarse.
- **puerto**: el número de puerto para la conexión.
- **usuario**: el nombre del usuario con el que registrarse en el servidor.
- **canal**: el nombre del canal al que conectarse.

Ejercicio 1: Funcionalidad Mínima

- Iniciar una sesión en el servidor.
- Unirse a una sala determinada.
- Enviar y recibir mensajes.
- Finalizar la sesión iniciada.

Ejercicio 1: Secuencia de acciones

- 1 Abrir un socket para conectarse.
- 2 Establecer una conversación con el servidor utilizando el protocolo IRC.
- 3 Esperar la recepción de comandos por parte del usuario o de mensajes IRC por parte del servidor.
 - 1 En caso de recibir un comando del usuario enviar al servidor el mensaje IRC correspondiente.
 - 2 En caso de recibir un mensaje IRC del servidor realizar la acción asociada dicho mensaje.
- 4 Termina la conexión con el servidor utilizando el protocolo IRC, cierra el socket y sale con código de respuesta 0.

Ejercicio 1: Interfaz de texto

- El programa leerá de teclado los comandos solicitados por el usuario local y mostrará en pantalla los mensajes del chat (públicos y privados) recibidos del servidor.
- Cada vez que el cliente reciba una solicitud de PING mostrará en pantalla el texto “- IRC: Recibido ping del servidor”.
- La lista de comandos soportados por el cliente será:
 - `<mensaje><CR>` - Mandará el texto `¡mensaje!` como mensaje público a toda la sala.
 - `:m <usuario> <mensaje><CR>` - Mandará el texto `<mensaje>` como mensaje público a toda la sala.
 - `:s<CR>` - Cerrará la conexión con el servidor y saldrá de la aplicación.
 - ...

Ejercicio 1: Interfaz Gráfica

- El botón connect realizará la misma funcionalidad que el inicio del programa.
- El botón send envía un mensaje con el formato de la interfaz de texto.
- Desarrollar chatConnect para la conexión y chatSend para el envío.

Ejercicio 1: No olvidar

- El programa deberá salvar en el fichero “traza.log” la secuencia completa de mensajes IRC intercambiados entre el cliente y el servidor.
- Se puede usar un formato parecido al de las trazas del enunciado.

Ejercicio 1: Valoración

- Puntuación máxima: 6.
- Se tendrá en cuenta:
 - Correcto funcionamiento de todos los casos.
 - La calidad de la memoria entregada y la claridad de las explicaciones que contiene (una mala documentación tendrá una influencia negativa en la nota).

Ejercicio 2: Chat de audio UDP

- Chat de voz por UDP.
- Se realizará un único programa cliente/servidor.
- Dependiendo de la invocación cambiará el comportamiento.

Invocar al programa

- Servidor:

- `voiceChat -s <puerto>`

- Cliente:

- `voiceChat -c <ip> <puerto>`

Objetivos

- Identificar las limitaciones de la comunicación (no corregirlas).
- Proponer soluciones posibles.
- Las conclusiones deben incluir un análisis brevemente detallado sobre cómo influyen el formato y las propiedades de la codificación (muestreo, calidad, etc) en la eficiencia de las comunicaciones utilizadas, es decir, lo bueno o malo que es el protocolo y si se necesitaría algún tipo de compresión para enviar la información.

Calificación

- Correcto funcionamiento: 2.
- Tipo de pruebas: 2.
- Una memoria breve, concisa, clara y directa (como la vida misma).