



PROGRAMACIÓN DE SERVICIOS Y PROCESOS

Dpto. INFORMÁTICA

1º Enviar una cadena de texto del cliente al servidor y leer la respuesta. Repetir el ejercicio encriptando de algún modo la cadena.

2º Implementa un cliente que accede al servicio UNIX "fecha y hora". El servidor concreto al que se conecta es al "localhost". El servicio "fecha y hora", por convenio, siempre está en el puerto 13. El software del servidor deberá ejecutarse continuamente en la máquina remota, esperando cualquier tráfico de red que "hable con él" en el puerto 13. Cuando el Sistema Operativo de este servidor recupera un paquete de red que contiene una petición para conectar con el puerto 13, activa el servicio de escucha del servidor y establece la conexión, que permanece activa hasta que es finalizada por alguna de las dos partes.

3º Implementación un cliente que accede al servicio UNIX "eco". El servicio "eco", por convenio, siempre está en el puerto 7. Aunque con frecuencia por razones de seguridad está cerrado.

4º Implementar un servidor de "eco" como el que proporcionan los servidores Unix en el puerto 7. Diseña un servidor concurrente, para ello:

- Se ha de especificar en la línea de comandos el número de puerto en el que el servidor acepta conexiones (por ejemplo, 8180). Si no se especifica nada, el número de puerto por defecto será el 7.
- El servidor se ha de quedar esperando las solicitudes de conexión de los clientes. Utilizando la clase `BufferedReader` junto con la clase `InputStreamReader` se ha de abrir un flujo de entrada desde el socket servidor. Con la clase `PrintWriter` junto con la clase `OutputStreamWriter` abrir un flujo de salida al socket. El programa ha de actuar como repetidor, recogiendo las líneas que llegan por el canal de entrada y escribiéndolas en el canal de salida, hasta que el usuario le indique que ha terminado, escribiendo un punto ".". Cuando ya no haya más líneas que leer, se recibirá un punto, lo cual hará que el servidor salga del bucle de entrada y cierre el socket servidor.
- Para probar que funciona, ejecute el programa servidor en una consola, y en otra terminal escriba: `telnet 127.0.0.1 8180`.

6º El código de un sistema cliente-servidor para la transmisión de archivos en cualquier formato. El Servidor tiene la capacidad de transmitir a los clientes que lo deseen un determinado archivo, por ejemplo `trabajo.ppt`, el cual se encuentra localizado en un path específico que el Servidor conoce. Entonces, uno o más clientes acceden a dicho servidor por medio de su dirección *ip* y colocan el *path* local en el que desean almacenar el archivo que están solicitando al servidor.

7º Crear un chat simple (muy simple), que usa Sockets para crear conexiones de red. Consta de un servidor (*Servidor.java*) y un cliente (*Cliente.java*), mediante los cuales se puede establecer una conversación. Si se escribe "TERMINAR" (sin comillas), la conexión se cierra.

8º Calcular el Cuadrado –TCP

En esta aplicación el cliente se conecta al servidor, para ello se debe introducir la dirección IP del servidor y los parámetros sobre los cuales el servidor debe hacer las operaciones. Una vez



PROGRAMACIÓN DE SERVICIOS Y PROCESOS

Dpto. INFORMÁTICA

conectado con el servidor este toma los parámetros y calcula su cuadrado y lo imprime por pantalla. Posteriormente envía los resultados al cliente, el cual también los imprime por pantalla.

En esta aplicación la conexión se realiza mediante conexión TCP, lo que permite al cliente y al servidor disponer de un stream que facilita una comunicación libre de errores.

El comportamiento para usar este tipo de socket es diferente en el cliente y el servidor. Cada uno de ellos utilizará unos métodos distintos. El esquema básico pasa por suponer que el servidor adoptará un papel pasivo y procederá a esperar conexiones de los posibles clientes. Mientras que los clientes serán los encargados de solicitar conexiones a los servidores de forma activa.

9º Calcular el Cuadrado –UDP

En esta aplicación el cliente envía un paquete al servidor, para ello se debe introducir la dirección IP del servidor y los parámetros sobre los cuales el servidor debe hacer las operaciones. Una vez enviado el paquete al servidor este toma los parámetros y calcula su cuadrado y lo imprime por pantalla. Posteriormente el servidor envía un paquete con los resultados al cliente con la dirección IP y el puerto del cliente obtenidos anteriormente, el cual también los imprime por pantalla.

En este caso se trata de un mecanismo más simple, puesto que el servicio sin conexión tan sólo nos ofrece un mero envío de datos. Puesto que no existe aquí la conexión no hay proceso previo alguno antes de enviar información. Para poder comunicar con otro proceso lo único que hay que hacer es crear el socket y utilizar sus métodos para el envío y recepción de información.