Práctica 2.2

TCP/IP. CLIENTE SERVIDOR

Jordi Blasco Lozano

Sistemas operativos y distribuidos

Grado en Inteligencia Artificial

## Indice:

[Indice: 2](#_Toc183099746)

[1. Introducción 3](#_Toc183099747)

[2. Trabajos a realizar 3](#_Toc183099748)

[3. Implementación del servidor 3](#_Toc183099749)

[3.1 Creación del socket y enlace 3](#_Toc183099750)

[3.2 Escucha y aceptación de conexiones 4](#_Toc183099751)

[3.3 Transferencia de archivos 4](#_Toc183099752)

[4. Implementación del cliente 4](#_Toc183099753)

[4.1 Creación del socket y conexión 4](#_Toc183099754)

[4.2 Recepción de archivos 4](#_Toc183099755)

[5. Resultados 5](#_Toc183099756)

## Introducción

En esta práctica se ha implementado una aplicación cliente-servidor utilizando sockets TCP. El objetivo principal es que el servidor transfiera el archivo "Google.html" al cliente, quien posteriormente lo mostrará por pantalla. El servidor se mantiene a la escucha por el puerto 9999, aceptando conexiones y generando un proceso hijo para cada cliente que se conecte.

## Trabajos a realizar

* Crear un servidor que pueda recibir conexiones y enviar un archivo al cliente.
* Crear un cliente que se conecte al servidor y reciba el archivo.

## Implementación del servidor

La función principal del servidor es escuchar a los clientes y transferirles el archivo.

## 3.1 Creación del socket y enlace

Primero, se usa la función **socket**() para crear un socket. Luego se configura la dirección del servidor usando la estructura **sockaddr\_in**y se conecta al socket con **bind**(). Después, usamos **listen**() para permitir que el servidor escuche hasta 5 conexiones.

## 3.2 Escucha y aceptación de conexiones

Con **accept**(), el servidor acepta conexiones de los clientes. Cada vez que se acepta una conexión, se crea un proceso hijo usando **fork**(). El proceso hijo se encarga de enviar el archivo, mientras que el proceso padre sigue esperando nuevas conexiones.

## 3.3 Transferencia de archivos

El archivo "Google.html" se abre y se lee en partes de tamaño BUFFERSIZE que se envían al cliente con **send**(). Cuando se termina de enviar el archivo, el proceso hijo cierra la conexión.

## Implementación del cliente

El cliente tiene la funcion de conectarse al servidor, recibir el archivo y posteriormente escribirlo.

## 4.1 Creación del socket y conexión

El cliente usa **socket**() para crear el socket y luego **connect**() para conectarse al servidor. Para ejecutar el cliente, se necesita especificar la dirección IP del servidor que sera pasada por parametro.

## 4.2 Recepción de archivos

Una vez conectado, el cliente recibe el archivo en partes usando **recv**(). El contenido del archivo se imprime en pantalla para asegurarse de que todo se recibió correctamente.

## Resultados

