

PROYECTO FINAL

DESCRIPCIÓN

Implementarás un sencillo servidor de archivos y un cliente para el mismo. Ser un repositorio de archivos es uno de los servicios más comunes que tiene un servidor Unix. Este servicio se puede realizar utilizando muchos tipos de protocolos como FTP, NFS, SMB, etc. Estos protocolos no son muy fáciles de implementar, porque lo que nos abocaremos a desarrollar nuestro protocolo propietario.

MODALIDAD

Individual

INSTRUCCIONES

ARRANQUE

Inicialmente, al servidor de archivos se le da un subdirectorio que se considerará como "directorio raíz". Cualquier ruta (path) que el cliente le dé al servidor se considerará relativo a este "directorio raíz". Por ejemplo, si el subdirectorio en el servidor es `/my_tiny_ftp/archivos`, y el cliente pregunta por el siguiente archivo `/data/config.txt`, el servidor le enviará el archivo `//my_tiny_ftp/archivos/data/config.txt`.

Como se comentó en un inicio, nosotros definiremos un protocolo propietario entre el cliente y el servidor. El cliente mandará comandos (como parámetros opcionales) y el servidor responderá. Cada mensaje transferido será precedido por el tamaño del mensaje, de forma que lado que recibe siempre sepa cuando bytes debe esperar. Un mensaje enviado por el servidor siempre contendrá un código de retorno y, quizás, contenga la salida de un comando (como el listado de un directorio) o el contenido de un archivo (para el caso de una petición de transferencia).

PROTOCOLO

Cada mensaje transferido contendrá tres campos (**en ese preciso orden**):

- Código del mensaje - 2 bytes.
- Longitud del mensaje - 4 bytes.
- Datos adicionales - Longitud variable.

Un cliente inicia una conexión con el servidor. El servidor manda una respuesta de bienvenida (código '1', con un "hola". El cliente puede mandar dos posibles mensajes:

- Un mensaje de envío de archivo (código '101' y la ruta del archivo).
- Un mensaje de desplegado de un directorio (código '102' y la ruta del directorio).
- Un mensaje de fin de conexión (código '103' con un "adios").

El servidor puede responder lo siguiente:

- Permiso denegado (código '201' y el mensaje de error).
- Archivo no encontrado (código '202' y el mensaje de error).

- Error interno (código '203' y el error generado).
- Comando no conocido (código '204' y el mensaje de error).
- La ruta es un directorio (código '205' y el mensaje de error).
- Enviando archivo (código '301' y el contenido del archivo).
- Directorio no encontrado (código '206' y el mensaje de error).
- La ruta no es un directorio (código '207' y el mensaje de error).
- Enviando el contenido de un directorio (código '302' y el listado del directorio).

Debes definir una constante para cada código de mensaje en un archivo header que sera usado tanto por el código del cliente y el servidor.

MEDIDAS DE SEGURIDAD

El servidor debe revisar cualquier mensaje que reciba el cliente:

- Para asegurarse que toda ruta comienza con el carácter '/' y NO contiene una secuencia de ".." en ella.
- Crear "buffers" lo suficientemente grandes para aceptar los comandos enviados. El cliente especifica un tamaño de mensaje demasiado grande, el servidor enviará un mensaje de "Error Interno" y termina la conexión.

ARCHIVO DE REGISTRO

El servidor debe escribir un mensaje de una línea en un archivo de registro (log file) registrando las siguientes operaciones (con tiempo, fecha y dirección ip de la conexión al inicio de cada mensaje):

- Cuando se ha establecido una conexión con un cliente.
- Cualquier comando que se reciba de un cliente. Adicionalmente se guarda el comando y parámetro recibido.
- Cualquier respuesta enviada hacia el cliente. Adicionalmente se guarda el mensaje enviado y, dependiendo del tipo de mensaje, el mensaje de error o la ruta del archivo.