

# Proyecto 2: Web Server

Eric López Tornas C-311

## Web Server Ejecución

Para compilar:

```
gcc webserver.c -o webserver
```

Para montar nuestro webserver en el puerto  $\langle port \rangle$  y a partir de la carpeta  $\langle folder \rangle$  escribimos:

```
./webserver  $\langle port \rangle$   $\langle folder \rangle$ 
```

## Iniciando Conexión

Cuando se ejecuta el servidor luego de comprobar que sean válidos los parámetros de entrada se crea el socket por el que el servidor va a esperar a que se conecten los clientes, esto se logra precisamente haciendo uso de la función *openlistenfd* la cual te devuelve el file descriptor del servidor.

Luego de crear el socket se configura el servidor usando *setsockopt* para que este termine y se reinicie automáticamente, se le asigna una dirección (y el puerto) mediante la función *bind* y finalmente se le indica al file descriptor del servidor que es un listening descriptor mediante la función *listen*.

Una vez que el socket del servidor está creado y esperando a que se conecte algún cliente se inicia un bucle en el cual se acepta la conexión entrante con el método *accept*, se lee el request del file descriptor del cliente, se parsea la uri para decodificar los caracteres de codificación porcentual, se responde la petición y se cierra el file descriptor del cliente.

## Peticiones y Respuestas

El servidor muestra inicialmente las carpetas y archivos que se encuentran en el serving directory y permite navegar entre las diferentes carpetas.

El servidor responde a peticiones de tipo GET, los restantes métodos no están implementados y por tanto producirán una respuesta de "Error 501 Not Implemented ", las request de tipo GET se tienen cuando se hace click sobre el enlace de alguna carpeta o de algún archivo, en el primer caso la respuesta del servidor será una página web (documento HTML) con los subdirectorios del directorio seleccionado, en el caso que cliqueemos un enlace a un archivo este se descargará en el dispositivo del cliente.

Para leer las peticiones del usuario se utiliza la función *readrequest* y para escribir la respuesta del servidor se emplea la función *write* y manejan el caso de interrupción de lectura/escritura por error.

Para identificar si la uri de la petición corresponde a una carpeta o a un archivo utilizamos el método *stat* para obtener las propiedades de la ruta y luego S\_ISDIR, si *stat* devolviese un valor menor que 0 es porque la ruta de la uri no se encuentra, a estas peticiones se les responde con "Error 404 Not Found".

En caso en que el S\_ISDIR devuelva 0 entonces estamos en presencia de una petición de descarga, para enviar un archivo al file descriptor del cliente se utiliza la función *sendfile*, indicando que se va a realizar una descarga, se escribe el encabezado en el file descriptor del cliente y luego se manda el archivo con *sendfile*.

## Funcionalidades Implementadas

Se ha implementado las funcionalidades básicas:

1. Recibe los puerto y directorio
2. El servidor web crea una página web con el contenido del directorio.
3. Es capaz de procesar peticiones regulares de navegadores comunes.
5. El cliente, a través de la página web, puede acceder a un nuevo directorio con un solo click.
6. El cliente es capaz de descargar un archivo con solo dar un click en el nombre.
7. El servidor web puede atender correctamente a un cliente y manejar una sola petición a la vez.