

Orientación del Proyecto 3 (webserver)

El objetivo del proyecto es hacer un pequeño servidor *web* que permita listar directorios y archivos en una computadora.

Funcionalidades Básicas (3 puntos)

1. Al ejecutar el programa, se le deben pasar 2 parámetros
 - el **puerto** por donde va a escuchar
 - el **directorio raíz** desde donde se va a mostrar información

```
user@machine $ gcc webserver.c
user@machine $ ./a.out 4000 /home/pqr
Listening in port: 4000
Serving Directory: /home/pqr
```

Ejecución del webserver

2. El servidor web debe crear una página web con el contenido del directorio. Para la funcionalidad básica, solo es necesario mostrar el nombre de los archivos y directorios.

```
<html>
  <head>Directorio /home/pqr/Desktop</head>
  <body>
    <table>
      <tr>
        <th>Name</th>
        <th>Size</th>
        <th>Date</th>
      </tr>
      <tr>
        <td><a href="/Desktop/directorio1">directorio1</a></td>
        <td>0</td>
        <td>2017-01-20 10:46:34</td>
      </tr>
      <tr>
        <td><a href="/Desktop/directorio2">directorio2</a></td>
        <td>0</td>
        <td>2017-02-28 12:34:56</td>
      </tr>
      <tr>
        <td><a href="/Desktop/file3">file3</a></td>
        <td>3.8k</td>
        <td>2017-03-30 09:12:11</td>
      </tr>
      <tr>
        <td><a href="/Desktop/file4">file4</a></td>
        <td>4.7G</td>
        <td>2017-12-31 11:59:59</td>
      </tr>
    </table>
  </body>
</html>
```

Ejemplo de html a generar

3. No es objetivo del proyecto que el servidor implemente el protocolo *HTTP* completamente. Pero es necesario que sea capaz de entender las peticiones que recibirá de los navegadores *web* más comunes (*Chrome*, *Mozilla Firefox*, *Opera*, etc.).

4. Para saber como está conformada la petición que recibirá el servidor *web* implementado por ustedes se recomienda que en los primeros momentos del desarrollo, abran un navegador, pongan la dirección de su servidor *web* e impriman toda la información que reciban.

```
GET /Desktop HTTP/1.1
Host: localhost:4000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/96.0.4664.45 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Mode: navigate
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,es;q=0.8,ja;q=0.7
```

Ejemplo de petición hecha por un chrome a mi servidor web

5. El cliente, a través de la página *web*, debe ser capaz de “entrar” a un nuevo directorio con solo dar un click en el nombre del mismo.

6. El cliente debe ser capaz de descargar un archivo con solo dar un click en el nombre correspondiente en la página *web* generada por el *webserver* de ustedes.

7. El servidor *web* debe ser capaz de atender correctamente a un cliente y manejar una sola petición a la vez.

NOTAS:

- para leer el contenido de un directorio, considere usar las funciones `opendir()` y `readdir()`
- Hacer todos los puntos **del 1 al 7 equivale a 3 puntos** en el proyecto.

Funcionalidades Adicionales

8. Mostrar información adicional de los archivos y directorios, además del nombre (fecha de modificación, tamaño, permisos, etc.) **(1pt adicional)**

- Esta funcionalidad debe permitir ordenar por nombre, fecha, tamaño, etc.

9. Permitir múltiples peticiones de varios clientes simultáneamente **(1 pt adicional)**

- no se debería bloquear ni haber demoras si varios clientes hacen peticiones
- se recomienda usar `fork()` para esto.

10. usar `select()` para permitir varias peticiones desde un mismo cliente **(1 pt adicional)**

- para probar esta funcionalidad el usuario debe poder empezar a descargar un archivo grande. Y mientras se descarga poder seguir navegando por los directorios
- incluso puede darse el caso de que el usuario tenga varios archivos descargándose
- valore el uso de funciones como `sendfile()` o alguna equivalente