

Decisiones de Diseño

Servidor

Comando para compilar el servidor

```
gcc -o server server.c
```

luego para ejecutar se deberá poner lo siguiente en la terminal

```
./server
```

El servidor permanece en ejecución constante, esperando y respondiendo a solicitudes entrantes

Manejo de Solicitudes

El servidor espera recibir solicitudes de lectura y escritura de los clientes y procesa estas solicitudes adecuadamente identificando el tipo de solicitud basado en el código de operación recibido.

El nombre del archivo solicitado se extrae del paquete recibido.

Lectura y Escritura de Archivos

Lectura (GET):

El servidor abre el archivo solicitado en modo de lectura binaria .

Espera y verifica la recepción para cada bloque enviado.

Escritura (PUT):

El servidor abre el archivo en modo de escritura binaria .

Envía una confirmación para cada bloque recibido.

Manejo de Errores

Se han implementado mecanismos para manejar errores comunes(, como archivos no encontrados o problemas de red, para garantizar que las transferencias de archivos sean robustas y confiables

Si hay un error al escribir el archivo en el servidor, el archivo se cierra y se muestra un mensaje de error, al igual que si se encuentra un error al crear o vincular el socket, el programa se termina con un mensaje de error, y ocurre lo mismo cuando el socket no se puede crear o vincular.

Terminación de Transferencias

El servidor incluye la capacidad de detectar la finalización de una transferencia de archivo, asegurando que todos los datos se reciban completamente y que las transferencias exitosas se confirmen adecuadamente.

Cliente

primero para compilar el cliente se debe poner en una terminal

```
gcc -o client client.c
```

luego para ejecutar el cliente se debe utilizar el comando

```
./cliente <server_ip> <operation> <local_file_path> <remote_file_name>
```

server_ip: Dirección IP del servidor TFTP.

operation: 'get' para descargar un archivo desde el servidor o 'put' para subir un archivo al servidor.

local_file_path: Ruta del archivo local en la máquina cliente.

remote_file_name: Nombre del archivo en el servidor TFTP.

Operaciones de Archivo

GET (recepción de archivos): Implementa la lógica para recibir archivos desde el servidor TFTP.

Envía una solicitud de lectura y recibe bloques de datos del servidor.

Después de recibir cada bloque, envía una confirmación de recepción al servidor.

PUT (envío de archivos): Implementa la lógica para enviar archivos al servidor TFTP.

Envía una solicitud de escritura y, tras recibir la confirmación inicial, comienza a enviar bloques de datos al servidor.

Envía cada bloque de datos y espera una confirmación de recepción del servidor.

Terminación de Transferencias

Una vez que el archivo se ha transferido completamente, el cliente cierra el socket UDP.

Después de completar cualquier operación, el cliente se cierra para asegurar que los recursos se liberen adecuadamente.

Manejo de Errores

Se ha añadido un manejo básico de errores para asegurar que los paquetes incorrectos sean descartados y se mantenga la consistencia del protocolo. En caso de que la transferencia de datos falle, el cliente cierra el socket y el programa finaliza. Del mismo modo, si se encuentra un error al escribir el archivo local, el archivo se cierra, se muestra un mensaje de error y posteriormente se cierra el socket. Además, si se reciben datos no válidos, el cliente ignora el paquete defectuoso y espera el siguiente