

Contents

Protocolo HFTP	1
Home-made File Transfer Protocol (HFTP)	1
inicia el intercambio	2
Comandos:	2
Respuestas:	2
Comandos	2
get_file_listing	2
Respuesta:	2
get_metadata FILENAME	3
Respuesta:	3
get_slice FILENAME OFFSET SIZE	3
Comando:	3
Respuesta:	3
quit	3
Consideraciones	3
Manejo de Errores	3
0	4
100	4
101	4
199	4
200	4
201	4
202	4
203	4
código iniciado en 1	4
errores que inician con 2	5
línea de comandos	5
-d directory	5
-p port	5
Preguntas	5
Tarea Estrella	5

Protocolo HFTP

Home-made File Transfer Protocol (HFTP)

un protocolo de transferencia de archivos casero, creado por nosotros específicamente para este laboratorio.

HFTP es un protocolo de capa de aplicación que usa TCP como protocolo de transporte. Un servidor de HFTP escucha pedidos en el **puerto TCP 19500**.

inicia el intercambio

Inicia el intercambio de mensajes mediante pedidos o **comandos** al servidor.

El servidor envía una **respuesta** a cada uno antes de procesar el siguiente hasta que el cliente envía un comando de fin de conexión.

En caso de que el cliente envíe varios **pedidos consecutivos**, el servidor HFTP los responde en el orden en que se enviaron.

es un protocolo ASCII, no binario,

Comandos:

consisten en una cadena de caracteres compuesta por elementos separados por un único espacio y terminadas con un fin de línea estilo DOS (`\r\n`)

El primer elemento define el tipo de acción y los elementos que siguen son argumentos

Respuestas:

comienzan con una cadena terminada en `\r\n`, y pueden tener una continuación dependiendo el comando que las origina.

Comandos

get_file_listing

no recibe argumentos y busca obtener la lista de archivos que están actualmente disponibles.

Respuesta:

0 OK\r\n archivo1.txt\r\n archivo2.jpg\r\n \r\n

recibe un argumento `FILENAME` especificando un nombre de archivo

get_metadata FILENAME

El servidor responde con una cadena indicando su valor en bytes.

Respuesta:

0 OK\r\n 3199\r\n

get_slice FILENAME OFFSET SIZE

recibe en el argumento FILENAME el nombre de archivo

La parte se especifica con un OFFSET (byte de inicio) y un SIZE (tamaño de la parte esperada, en bytes), ambos no negativos

El servidor responde con el fragmento de archivo pedido codificado en base64 y un \r\n.

Comando:

get_slice archivo.txt 5 20

Respuesta:

0 OK\r\n Y2Fsb3IgcXVIIGhhY2UgaG95LCA=\r\n4

quit

Este comando no recibe argumentos y busca terminar la conexión. El servidor responde con un resultado exitoso (0 OK) y luego cierra la conexión.

Consideraciones

- Los nombres de archivos no deberán contener espacios,
- Atención que algunos caracteres fuera del lenguaje Inglés se representan con dos Bytes.

Manejo de Errores

En caso de algún error, el servidor responderá con códigos de respuestas diferentes a 0, más algún texto descriptivo a definir por el implementador.

0

La operación se realizó con éxito.

100

Se encontró un carácter `\n` fuera de un terminador de pedido `\r\n`.

101

Alguna malformación del pedido impidió procesarlo

199

El servidor tuvo algún fallo interno al intentar procesar el pedido.

200

El comando no está en la lista de comandos aceptados.

201

La cantidad de argumentos no corresponde o no tienen la forma correcta.

202

El pedido se refiere a un archivo inexistente.

203

El pedido se refiere a una posición inexistente en un archivo . 6

código iniciado en 1

son considerados fatales y derivan en el cierre de la conexión una vez reportados por el servidor.

errores que inician con 2

permiten continuar con la conexión y recibir pedidos posteriores.

Se aplica particularmente al comando y debe generarse cuando no se cumple la condición **OFFSET + SIZE ≤ filesize**.

línea de comandos

-d directory

para indicarle donde están los archivos que va a publicar.

-p port

para indicarle en que puerto escuchar. Si se omite usará el valor por defecto.

Preguntas

1. ¿Qué estrategias existen para poder implementar este mismo servidor pero con capacidad de atender múltiples clientes simultáneamente? Investigue y responda brevemente qué cambios serían necesario en el diseño del código.
2. Pruebe ejecutar el servidor en una máquina del laboratorio, mientras utiliza el cliente desde otra, hacia la ip de la máquina servidor. ¿Qué diferencia hay si se corre el servidor desde la IP “localhost”, “127.0.0.1” o la ip “0.0.0.0”?

Tarea Estrella

En caso de implementar el servidor con capacidad de atender múltiples clientes simultáneamente, se otorgarán puntos extras.