Concurrent File Transfer TCP Server - How It Works

The concurrent TCP server (listening to the port specified as the first parameter of the command line, as a decimal integer) after having established a TCP connection with a client, accepts file transfer requests from the client and sends the requested files back to the client. The server create processes on demand (a new process for each new TCP connection).

The files available for being sent by the server are the ones accessible in the server file system from the working directory of the server.

The client can connect to a TCP server (to the address and port number specified as first and second command-line parameters, respectively). After having established the connection, the client requests the transfer of the files whose names are specified on the command line as third and subsequent parameters, and stores them locally in its working directory. After having transferred and saved locally a file, the client print a message to the standard output about the performed file transfer, including the file name, followed by the file size (in bytes, as a decimal number) and timestamp of last modification (as a decimal number).

The protocol for file transfer works as follows: to request a file the client sends to the server the three ASCII characters “GET” followed by the ASCII space character and the ASCII characters of the file name, terminated by the ASCII carriage return (CR) and line feed (LF):

(Note: the command includes a total of 6 characters plus the characters of the file name). The server replies by sending:

Note that this message is composed of 5 characters followed by the number of bytes of the requested file (a 32-bit unsigned integer in network byte order - bytes B1 B2 B3 B4 in the figure), then by the timestamp of the last file modification (Unix time, i.e. number of seconds since the start of epoch, represented as a 32-bit unsigned integer in network byte order - bytes T1 T2 T3 T4 in the figure) and then by the bytes of the requested file.

The client can request more files using the same TCP connection, by sending many GET commands, one after the other. When it intends to terminate the communication it sends:

Q	U	I	T	CR	LF

(6 characters) and then it closes the communication channel.
In case of error (e.g. illegal command, non-existing file) the server always replies with:

-	E	R	R	CR	LF

(6 characters) and then it closes the connection with the client.

The code can be compiled with the following commands, issued from the source folder:

gcc -std=gnu99 -o server server2/*.c *.c -Iserver2 -lpthread -lm

gcc -std=gnu99 -o client client1/*.c *.c -Iclient1 -lpthread -lm