Documentation

Usage

- Type "*make all*" to compile the code.
- Run the server using "*./server* port" - The port must be an integer between 1024 and 65535 inclusive.
- The server handles both IPv4 and IPv6.
- The server accepts all "GET" requests and returns a status code depending on the result of the request:
    - Successful requests will return status code 200 OK followed by the requested file or page.
    - Other requests will return status code 501 Not Implemented.
- The server will return a status code followed by a content length header. The content length header states the length of the body of the response, whether this response is HTML code or a file.
- Any errors produced by the code will be printed into the *stderr* stream.

Status codes

200 OK - The request was correctly received and the response was generated without any errors.
404 File Not Found - The request was correctly received but the file requested was not found on the server.
500 Internal Server Error - The request was correctly received but the server encountered an error. Depending on the request, this means that either the server failed to read a file or that the server failed to open a directory.
501 Not Implemented - The type of request is not supported on the server. The server only supports "GET" requests.

Included files

1558190/doc/documentation.pdf
1558190/doc/diary.pdf
1558190/src/Makefile
1558190/src/server_multi.c
1558190/src/service_client_socket.c
1558190/src/service_client_socket.h
1558190/src/make_printable_addr.c
1558190/src/make_printable_addr.h

Code

server_multi.c

main:

- Check the given arguments
- Get a socket to use socket
- Bind to the socket
- Listen for connections to the socket
- Accept any incoming connections and handle them with a new thread (client_thread)

client_thread:

- Create a printable version of the IPv4 or IPv6 address
- Pass relevant information to service_client_socket where the connection is handled

make_printable_addr.c

- Returns a printable string version of an IPv4 or IPv6 address

service_client_socket.c

service_client_socket:

- Read data sent from the client
- Parse the HTTP request into request method (e.g. “GET”), host (e.g. localhost:8088), and resource location (e.g. /test.txt)
- If the resource is a file, use read_file to get the file contents and then use send_file to send the file to the clien
- If the resource is a directory, look inside the directory and return its contents to the client, along with links to each file and subdirectory

read_file:

- Copies the contents and length of the given file path into the given file buffer and length arguments

send_response:

- Sends an HTTP message consisting of a header and body to the client

send_file:

- Sends a file over HTTP with a header to the client