Web Server Documentation

The program is a simple c server that can process GET and HEAD requests. It can serve a range of different files type requests such as .html pages and .png images. In the src directory contains a Makefile that will compile the program by typing 'make' or 'make all' into the terminal. The program can then be run by typing './webserver <port number>'.

Threading
For each thread created, it uses a structure that contains all the relevant information such as the socket the client is connected on, the client's address and the size of this address. The structure is then freed and the thread terminated after client has been served their requested web page.

File Types
There is also a structure used to identify the types of files that the program can serve and for use in the response header 'Content type: '.

Functions
*void error(char *msg)*
This function prints an error message.

*void write_new(int sock, char *msg)*
A helper function that allows simple message publishing to a socket.

*int main(int argc, char *argv[])*
The main function of the program accepts one argument, the port number to which the server will be listening on. It will then attempt to open a socket on this port, bind it to an address and then will listen for connections. For each new request, a new thread is made to deal with the request.

*void *handle(void *param)*
This function is called every time a new request is received and takes one parameter, the thread control block. This function will read from the request and place it into a buffer. To find the request type, we ignore the headers and place a terminating character after the file name stored in the buffer. It will then check if a null URI is given, and change it to point to the index if this is the case. The head of the buffer is then checked to see what type of request it is, this web server can only process HEAD and GET requests.
For GET requests, the type of file to be served is determined by looking at the extension characters of the file name. The file is then opened, if it is not found then an error message would be displayed server side and the content will not be sent. If the file was successfully opened, then the server will send a response containing the OK message and the content type of the file being served. The file is then written to the buffer to be sent to the client.
For HEAD requests, the process is the same as GET, but the file is not sent to the client at the end of the process and the content length is instead given.
If the request was not recognised due to bad syntax or an unimplemented feature, then the response message “404 Not Found” and the error html web page will be sent regardless of which reason.
After a request is processed, the thread control block is freed from memory, the connection is closed with the client and the thread terminated.

int get_file_size(int fd)
This function is used to get the size of a file to be sent.