GET request

TABLE OF CONTENTS

Files

Implement `handle_files_request` to handle HTTP `GET` requests for files. You will need to call `serve_file` accordingly. You should also be able to handle requests to files in subdirectories of the files directory (e.g. `GET /images/hero.jpg`).

- If the file denoted by path exists, call `serve_file` on it. Read the contents of the file and write it to the client socket.
    - Make sure you set the correct `Content-Length` HTTP header. The value of this header should be the size of the HTTP response body, measured in bytes. For example, `Content-Length: 7810`. You can use `snprintf` to convert an integer into a string.
    - You must use the `read` and `write` syscalls for this assignment. Any implementations using `fread` or `fwrite` will not earn any credit. This is purely for pedagogical reasons; we want you to be comfortable with the fact that low-level I/O may or may not perform the entire operation on all the bytes requested.
- Else, serve a `404 Not Found` response (the HTTP body is optional) to the client. There are many things that can go wrong during an HTTP request, but we only expect you to support the `404 Not Found` error message for a non-existent file. After finishing this part, curling for `index.html` should output the contents of the file `index.html`.

Directories

Implement `handle_files_request` to handle HTTP `GET` requests for both files and directories.

- You will now need to determine if `path` in `handle_files_request` refers to a file or a directory. The `stat` syscall and the `S_ISDIR` or `S_ISREG` macros will be useful for this purpose. After finding out if `path` is a file or a directory, you will need to call `serve_file` or `serve_directory` accordingly.

- If the directory contains an `index.html` file, respond with a `200 OK` and the full contents of the `index.html` file. You may not assume that directory requests will have a trailing slash in the query string.
  - The `http_format_index` function in `libhttp.c` may be useful.
- If the directory does not contain an `index.html` file, respond with an HTML page containing links to all of the immediate children of the directory (similar to `ls -1`), **as well as a link to the parent directory**.
  - The `http_format_href` function in `libhttp.c` may be useful.
  - To list the contents of a directory, good functions to use are `opendir` and `readdir`.
- If the directory does not exist, serve a `404 Not Found` response to the client.
- You don’t need to worry about extra slashes in your links (e.g. `//files///a.jpg` is perfectly fine). Both the file system and your web browser are tolerant of it.
- You do not need to handle file system objects other than files and directories (i.e. you do not need to handle symbolic links, pipes, or special files).
- Remember to close the client socket before returning from the `handle_files_request` function.
- Make helper functions to reuse similar code when you can. It will make your code easier to debug!

After finishing this part, curling for the root directory `/` should output the contents of the file `index.html`. All tests for Basic Server tests should pass on the autograder.
