TCP Programming with Sockets - Rudimentary HTTP Server

Purpose

To practice socket programming with TCP/IP, including the system calls that were learned previously, as well as `bind`, `listen`, and `accept`.

Assignment

Write a C++ program that implements a basic HTTP server. If this is done properly, you should be able to connect to it through a web browser, but this will not be one of the criteria for grading.

The program will essentially be a loop that goes on forever (until it is killed), waiting for a client to connect. When a client connects, it accepts that connection, calls `fork` to make a child process, and handles communication from that client in the child process. The parent process should continue to wait for more connections, accepting them and forking as necessary. The program will accept two command line parameters:

1. the port number on which the server will be listening
2. the path to a directory that will serve as the root directory of the web server

For example:

```
% ./z123456 9001 www
```

The requests received by the program will be of the form:

```
GET path
```

where the *path* is the path, relative to the directory specified as the second command line parameter, of the file that is being requested. There are several rules on what can constitute a valid path, and they are as follows:

- it must begin with a "`/`"
- it may contain additional "`/`" path separators to access subdirectories
- a single "`/`" character refers to the directory specified in command line
- a trailing "`/`" in the pathname can be ignored if the path refers to a directory
- any data in the request beyond the path should be ignored
- it may not contain the substring "`..`"

If the path refers to a directory, then:

- if the file called "`index.html`" exists in that directory, send the contents of that file to the client
- if not, send a list of the files in the specified directory to the client (do not include files that start with "`.`")

If the path refers to a file, then the content of the file should be sent to the client.

Error Checking

If the command line arguments are incomplete, or if the path to the web server's root is invalid, print an error message to `stderr` and exit with an error code. If any of the system calls fail, the program should use `perror` to report what happened and exit with an error code. If the path in the GET request is invalid, or if a file or directory cannot be accessed, then an appropriate error message should be sent to the client to notify them.

Other Notes

- This is a simple TCP server. If you need a tool to test it, you can use the `telnet` command to connect to your server at the specified port. Type your command and hit enter, and the client should display what the server returns.

  ```
  % telnet localhost 9001
  Connected to localhost.
  Escape character is '^]'.
  GET /
  fileOne.html fileTwo.html
  Connection closed by foreign host.

  % telnet localhost 9001
  Connected to localhost.
  Escape character is '^]'.
  GET fileOne
  Error: fileOne not found
  Connection closed by foreign host.

  % telnet localhost 9001
  Connected to localhost.
  Escape character is '^]'.
  GET fileOne.html
  ... contents of file ...
  Connection closed by foreign host.
  ```

- There will be source code for a basic TCP client posted on BlackBoard. If you want, this can also be used as a test.
- Make sure that the assignment is contained in a single file called "`z123456.cxx`", replacing z123456 with your actual z-id.
- Make sure that your program compiles, links, and runs properly on turing/hopper.
- If you'd like to continue working on this after the requirements here, you can learn more about the rules for how a web server should behave are available at the URL `https://tools.ietf.org/html/rfc2616`.

Submission

Submit your C++ source code via BlackBoard.