Aidan Smith

CSE 130

Writeup for Assignment 1

Testing:

In order to thoroughly test my program, I tested the functionality of the three different calls, with varying files and what they contained. I used the flag -v to check that the headers matched what was expected. I made sure to compile with all the flags so that no unforeseen errors arose.

Questions:

What fraction of your design and code are there to handle errors properly? How much of your time was spent ensuring that the server behaves “reasonably” in the face of errors?

Probably a third of my overall code is dedicated to catching errors and disconnecting clients. Whilst writing code I would brainstorm where a server-client connection could run into an issue. I made sure that it would keep the server on at all costs, even ditching the client if need be. I spend about half of my time thinking about and writing code for errors.

List the “errors” in a request message that your server must handle. What response code are you returning for each error?

The server must check and handle errors in the request message that are:

- File name is not composed of alphanumeric or - or _ characters. These would be an error code 400, client’s fault.
- HTTP version must be checked. If not 1.1 then error code 400 is given, as it is also the fault of the client.

- If the method is not GET, PUT, or HEAD then it had to error with code 400, still the client's fault.
- If no file name is provided, we cannot read from the terminal so it is an error 400.
- If the file cannot be opened due to permissions then it is an error 403, lacking permission.
- If the file cannot be found, then it is an error 404, not found.
- Most other errors that could occur would be some server error, so the rest of the errors are labeled as 500, just generically if there is an error.

What happens in your implementation if, during a PUT with a Content-Length, the connection is closed, ending the communication early?

The socket would not be read so it would quit the function and would error. It wouldn't be able to print anything without a client so the connection would end and open up for a new client.

Does endianness matter for the HTTP protocol? Why or why not?

Yes! It reads in the request in order to ensure the information is processed properly. If it were out of order then the functions would read information incorrectly and all hell could break loose. Likely an error would be triggered to avoid this, but if endianness is ensured then those errors can be avoided.