Aaron Nguyen
1585632
Assignment 1 Writeup

Testing

For this lab the majority of the testing involved unit testing each of the functions and using telnet and curl to test my code. Telnet a linux command was used in the initial process of the code to test the activation and connection of a server to a client. Telnet was also used to test how to sending and receiving messages worked in c++ and with sockets. Curl was used in the later stages of the program test GET and PUT header processing and responses in the server. The unit were used on the internal functions of the main functions that took care of GET and PUT. Most of the time spent testing was with the Write() and Send() functions in c++ to perfect the outputs to the client.

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?

Around a quarter of my code is dedicated to handling errors. Most of my spent ensuring that the server behaves reasonably in the face of errors.

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

Bad Request, Forbidden, Not Found, and Internal Server Error, are the error messages that my server handles. In the same order the response codes are 400 (Bad Request), 403(Forbidden), 404(Not Found), and 500(Internal Server Error).

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

If the connection is closed early my program is writing as it reads the file. So whatever data has already been transferred is recorded into a file.

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

Endianness does matter in the HTTP protocol because in order for actions to take place certain commands or bytes need to passed in first to initiate any action. If it is not a header could be passed in after the data and when the server realizes it needs to write a file there be no data to be written after receiving the header bytes.