David John Neill
16331809

3D3 Project1 Report

Design:

In this project I decided not use to HTTP header and .cpp files but to have everything in the one file, I then call each function when needed. This may lead to project/code not being as efficient as others however in the specification it did not say that you had to make it efficient.
Firstly I designed the web-server to be able to take a request and then turn it into a response. I did this by using a desktop application called postman, this would generate an HTTP request and then I would build my code so that it could take in the request, process the information and then output the appropriate HTTP response and correct data from the file it was searching for. I also passed an argument into the web-server.cpp this allowed to pin point the location of where the file was saved in my hard drive. It meant than when looking for the file later on it would save time and effort.
Once I had this done and working correctly, I could then start working on the web-client .cpp. My web-client takes in an URL and then extracts the information that it needs. Once it has this information it then begins to compose an HTTP request. Once it has the request formed It then sends the request to server.
The web-client gets a response, acknowledges this and then gets the data/information from the HTTP response. It then opens up a file, wipes the file of any previous information and writes the new information to it and saves it.

Problems and Solutions:

1. For me the main problem was understanding the concepts of what the project was actually telling us to do. Eventually after many hours of research and talking to both friends and UGPC in trinity I knew what I had to do.
2. When displaying either the HTTP request or response, it would print out both of these at the same time, this caused me great confusion, however using the function memset() reset the buffers and it fixed the problem.
3. Sometimes when compiling the web-server.cpp I would get very strange and random errors so to fix this I would simply open up a new file and copy the code in and it would compile and run as usual.
4. At the beginning I was getting errors called “aborting” however this was an easy fix as all I had to do as close the socket.

Instructions on how to run:

See video attached within folder.

Testing Code:

With testing I tested it many different ways.

1. Simple non-persistent connection taking in one URL and outputting the information into a file.
2. Up grading the pervious statement by allowing it to have more than one URL input
3. Now testing the persistent connection with 1 URL and saying yes to Entering another URL
4. Now testing the persistent connection with more than 1 URL and saying yes to Entering another URL
5. Now testing the persistent connection with more than 1 URL and saying no to Entering another URL
6. Doing all the above steps again but saving information into a file.