CSS432

Terence Ho

February 9, 2019

**HW2: Simplified HTTP Retriever and Server Report**

Documentation

## Description

This is a simple HTTP retriever and server that works with any web browser. It is used to demonstrate the GET HTTP requests that web servers create as well as the responses that are passed.

## Retriever setup

The retriever takes in a domain or IP and port as well as the path. It then creates a socket to connect to the server. The retriever then sends a GET request for the file at the inputted path. If the response is HTTP 200 OK, then the content is displayed on the console and saved. If there is any other response code, the response code is displayed on the console and the content is not saved onto the file response file. Finally, when everything is done, the receiver closes the socket.

The retriever is split into three sections. The main method runs the retriever and calls on the startRequest() method in order to start a GET request. It also parses the IP or domain depending on which one it is using regular expression. The startRequest() method is used to start the receiver’s request. The Receiver() is used to setup the receiver and set the global variables. Most of the functions are done through the startRequest() where the response from the server request is recorded onto the received.txt file. If there is an error perror is used to display a response to the console.

## Server setup

The server creates a TCP socket. It then listens for a connection request. When the client connects to the socket, a new pthread is created. The server then reads the GET request from the client and sends a server response back. If everything was fine, it returns a HTTP 200 OK, if the file is not found it returns 404 Not Found. If the file was not authorized, it returns 401 Unauthorized. If the file was forbidden, it returns 403 Forbidden. If there was any other error, it returns a 400 Bad Request.

The server is also split into three parts. Similar to the retriever, there is a main method that calls the startServer() method. This then starts the TCP socket. The Server() sets all the global variables.

Execution

test.sh is used to test the server and build.sh is a script for compiling the code for the server and the retriever. It can be run by using “sudo bash test.sh” and “sudo bash build.sh”.



