HW2 - Simplified HTTP Retriever and Server

Re-submit Assignment

Due Apr 28 by 11:59pm **Points** 25 **Submitting** a file upload
File Types zip and tgz **Available** after Apr 3 at 12am

1. Overall Requirement

You will write two programs that exercise a simplified version of HTTP. **The retriever will work in conjunction with any web server and the server will work in conjunction with any web browser**. This way, you can test your software independently of each other.

- You need to write C++ implement the retriever and server.
- You should not use existing libraries to retrieve HTTP files or perform socket communication.
- You need to write both the build script and demo script files using bash shell script.

2. Detailed Requirement

- Retriever
  - Your retriever takes in an input from the command line and parses the server address and file (web page) that is being requested.
  - The program then issues a GET request to the server for the requested file.
  - When the file is returned by the server, the retriever outputs the file to the screen and saves the retrieved file to the file system.
  - If the server returns an error code instead of an OK code, then the retriever should not save the file and should display on the screen whatever error page was sent with the error.
  - Your retriever should exit after receiving the response.
  - The server name can be either an IP address for simplicity.
- Server
  - Your server waits for a connection and an HTTP GET request (Please perform multi-threaded handling).
  - Your server only needs to respond to HTTP GET request.
  - After receiving the GET request
    - If the file exists, the server opens the file that is requested and sends it (along with the HTTP 200 OK code, of course).
    - If the file requested does not exist, the server should return a 404 Not Found code along with a custom File Not Found page.
    - If the HTTP request is for SecretFile.html then the web server should return a 401 Unauthorized.

        - If the request is for a file that is above the directory structure where your web server is running (for example, "GET ../../../etc/passwd"), you should return a 403 Forbidden.
        - Finally, if your server cannot understand the request, return a 400 Bad Request.
    - After you handle the request, your server should return to waiting for the next request.
- Build script
    - You need to provide a build script to build your retriever and server.
- Test script
    - You need to create a script to test your retriever. This script should help you test your retriever when retrieving "authorized," "unauthorized," and "forbidden" files.

3. What to submit

You must submit the following deliverables in a zip file. If your code does not work in such a way that it could have possibly generated your results, you will not receive credit for your results. Some points in the documentation, evaluation, and discussion sections will be based on the overall professionalism of the document you turn in. You should make it look like something you are giving to your boss and not just a large block of unorganized text.

Your code should include both server and retriever, with a build and demo script. The demo script should run through all of the following test cases, together with screenshots of your programs executing all of the test cases. Make sure your screenshots are properly labeled!

1) Real Web browser accessing your server (screenshot only)

2) Your retriever accessing a real server (screenshot and demo script)

3) Your retriever accessing a file from your server (demo script)

4) Your retriever requesting an unauthorized file from your server (demo script)

5) Your retriever requesting a forbidden file from your server (demo script)

6) Your retriever requesting a non-existent file from your server (demo script)

7) Your retriever sending a malformed request to your server (demo script)

Criteria	**Percentage**
Documentation of your algorithm including explanations and illustrations in one or two pages	10
Source code that adheres good modularization, coding style, and an appropriate amount of comments. Retriever: 25 points, Server: 25 points, build script: 10 points, demo script and required screenshots: 20 points.	80

Execution output such as a snapshot of your display/windows. Or, submit partial contents of standard output redirected to a file. You don't have to print out all data. This does not include the required screenshots but all other outputs to help understand the execution and performance of your program.	10
Total	100

Your grade for this assignment will be "total points/4."

For reference: **http://tools.ietf.org/html/rfc1945** **(http://tools.ietf.org/html/rfc1945)**

Rubric

Criteria	Ratings		Pts
Server *Code for HTTP Server*	6.0 pts Full Marks	0.0 pts No Marks	6.0 pts
Retriever *Code for HTTP Retriever*	6.0 pts Full Marks	0.0 pts No Marks	6.0 pts
Screenshots	4.0 pts Full Marks	0.0 pts No Marks	4.0 pts
Documentation	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts
Execution Output	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts
Demo Script	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts
			Total Points: 25.0