CS118 Project 1 Webserver

First Student Name: Yichen Wu
UID: 504294181
Seasnet Login Name: yichenw

Second Student Name: Kaiwen Huang
UID: 204171803

High level Description of the Server Design

The server makes connections with clients, listens to their requests and provide the requested file to the client.

Once the program receives a message from the client, it takes out the first line of the request and check if server support such kind of request.
If server does support this request, it tries to get the file requested.
Server also finds the file type of the requested file for it to be properly displayed by the client's browser.

Return:
If the request is not formatted correctly, server returns 400 BAD REQUEST.
If the request uses method not implemented by the server, server returns 501 BAD METHOD.
If the requested file/page is not found in the system, server returns 404 NOT FOUND.
If found, server returns 200 OK with the requested file.
Upon request with no specific file specified, server returns a default page.

Difficulties and solutions

We met some display syntax issues, but then we managed to find out the proper content-type needed for files to be properly displayed.
We also have no experience with html codes, but it's not hard to find examples.

Limitation

PDF files can be properly displayed by browser if the source code is compiled under and for Linux/Unix system. But if the source code is compiled into .exe under windows/cygwin, PDF file cannot be properly transferred and displayed to the client.

How to compile and run

We have written a Makefile to ease the compilation and running process.

1. Simply type 'make' and the executable 'webserver' will be generated
2. Then type: ./webserver <Port Number(e.g. 50000-59999)>
3. Open up a Browser and type in address bar: localhost:<Port Number> to see the front page.

You may also use command: 'make clean' to delete webserver and webserver.o .

Sample Outputs

```
$ ./webserver 50023
Here is the message:
GET / HTTP/1.1      This line is the only line that server uses to form a response.
Accept: text/html, application/xhtml+xml, image/jxr, */*
Accept-Language: en-US,en;q=0.8,zh-Hans-CN;q=0.7,zh-Hans;q=0.5,de-DE;q=0.3,de;q=0.2
User-Agent: Mozilla/5.0 (WindowsServer side: NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chr
```

Retrieving resource / Request is successful, and the default page will be sent to client.

----------next request---------- Process the next request (in a while loop)

$ ./client localhost 50023
Message Sent:
GET / HTTP/1.1 (Method Path Protocol)
Accept: text/html, application/xhtml+xml, image/jxr, */*
Accept-Language: en-US,en;q=0.8,zh-Hans-CN;q=0.7,zh-Hans;q=0.5,de-DE;q=0.3,de;q=0.2
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chr

Server's Response:
HTTP/1.1 200 OK (Request is successful. Now server is providing you…)
Content-type: text/html (…an html page…blahblahblah)

```
<!DOCTYPE html>
<html lang=en>
  <meta charset=utf-8>
  <meta name=viewport content='initial-scale=1, minimum-scale=1, width=device-width'>
  <title>Webserver</title>

  <p>This server provides you html, jpg, pdf, and jpeg files.</p>
  <p>You may try to access <a
href="http://localhost:50023/CS118_Project1.pdf">CS118_Project1.pdf</a></p>
  <p>You may try to access <a href="http://localhost:50023/README.txt">README.txt</a></p>
  <p>You may try to access <a href="http://localhost:50023/webserver.c">webserver.c</a></p>
  <p>You may try to access <a href="http://localhost:50023/favicon.ico">favicon.ico</a></p>
</body></html>
```