CISC 4615 Data Communication & Networks

Group Programming Assignment 3: Simple HTTP Server (Due: 12.7)

Introduction

In this project, you will need to implement a simple HTTP server.

Requirements

Your HTTP implementation will have four major pieces:

- Process the request from client (the browser)
- According to the request, send back the response
- Repose to five different requests (all of them are get methods)
  - Get index.html --- (GET http://localhost:9001/)
  - Get style.css ---- (GET http://localhost:9001/style.css)
  - Get app.js --- (GET http://localhost:9001/app.js)
  - Get fordham_logo.png --- (GET http://localhost:9001/fordham_logo.png)
  - Get FordhamUniversity.mp4 --- (GET http://localhost:9001/FordhamUniversity.mp4)
- Send corresponding status code, e.g. 404 and 400

Step 1: Process the client request

As we learned in the lecture, the follow block presents a sample request.

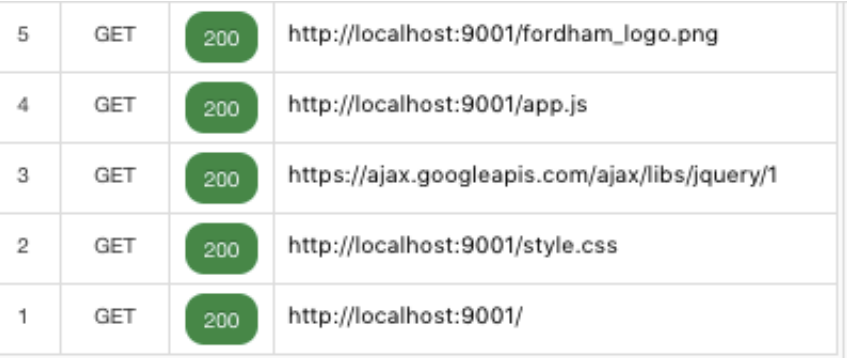
```
GET /style.css HTTP/1.1
Host: localhost:9001
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_0) AppleWebKit/537.36 (KHTML
, like Gecko) Chrome/70.0.3538.110 Safari/537.36
Accept: text/css,*/*;q=0.1
Referer: http://localhost:9001/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: _ga=GA1.1.858035700.1535167991; _xsrf=2|bd2adcfe|71d091e1f1202f43fddd64acca4e
b367|1540861896; username-localhost-8889="2|1:0|10:1541825212|23:username-localhost-8
889|44:NTI4MTI2ODczN2E4NDdjNmFjNDhiNGVhN2Q3ZTUwMDc=|f845692d430c24f4879daaf80e853185a
3250da4d5e968f42673c6d8f6b36e11"; username-localhost-8888="2|1:0|10:1542112177|23:use
rname-localhost-8888|44:NzViODhjNDZkOWM5NGFmYjhlMTIxOTY4ZTNiMDJjNGU=|1e00cc30e3f985f9
e112d56627f1d864d7abf38a56cee343ffbfed9f010697a4"
```

We have `status line`, `header lines`, and `cookie`, `data` fields.

In this project, you just need to process the `status line`, which is the first line of the request, `GET /style.css HTTP/1.1`. By processing this line, you code, your program needs to understand that it has to send the style.css file to the client (broswer).

Step 2: Construct the response

In this project, you will have five different files to send (GET mp4 video is not showed here),



5	GET	200	http://localhost:9001/fordham_logo.png
4	GET	200	http://localhost:9001/app.js
3	GET	200	https://ajax.googleapis.com/ajax/libs/jquery/1
2	GET	200	http://localhost:9001/style.css
1	GET	200	http://localhost:9001/

The GET 3 is sending to googleapis.com, so Google's server will handle it.

You will need to the `send a file` step in lab2 to read a file and load it into a buffer. Then, you can send this buffer to the client.

To lower the difficulty level, I copied the function from one of our classmates. I tested myself that you can use this function to implement your project.

```
int read_file_to_buffer(char* filename, char buffer[MAXLINE]) {
    size_t len;
    FILE *fp = fopen(filename, "r");
    if (fp != NULL) {
        len = fread(buffer, sizeof(char), MAXLINE-1, fp);
        if (ferror(fp) != 0) {
            return -1;
        } else {
            buffer[len] = '\0';
        }
    } else {
        return -1;
    }
    // Clean exit.
    fclose(fp);
    return len;
}
```

However, before you send this file, you have to send the status code and headers so that the client knows how to process it.

```
Status line:
        HTTP/1.1 200 OK
Response headers:
        Date: Wed, 10 Sep 2003 00:26:53 GMT
        Server: Apache/1.3.26 (Unix) PHP/4.2.2 mod_perl/1.27mod_ssl/2.8.10 OpenSSL/0.
9.6e
        Last-Modified: Tue, 09 Sep 2003 19:24:50 GMT
        ETag: "1c1ad5-1654-3f5e2902”
        Accept-Ranges: bytes
        Content-Length: 5716
        Keep-Alive: timeout=15, max=100
        Connection: Keep-Alive
        Content-Type: text/html
```

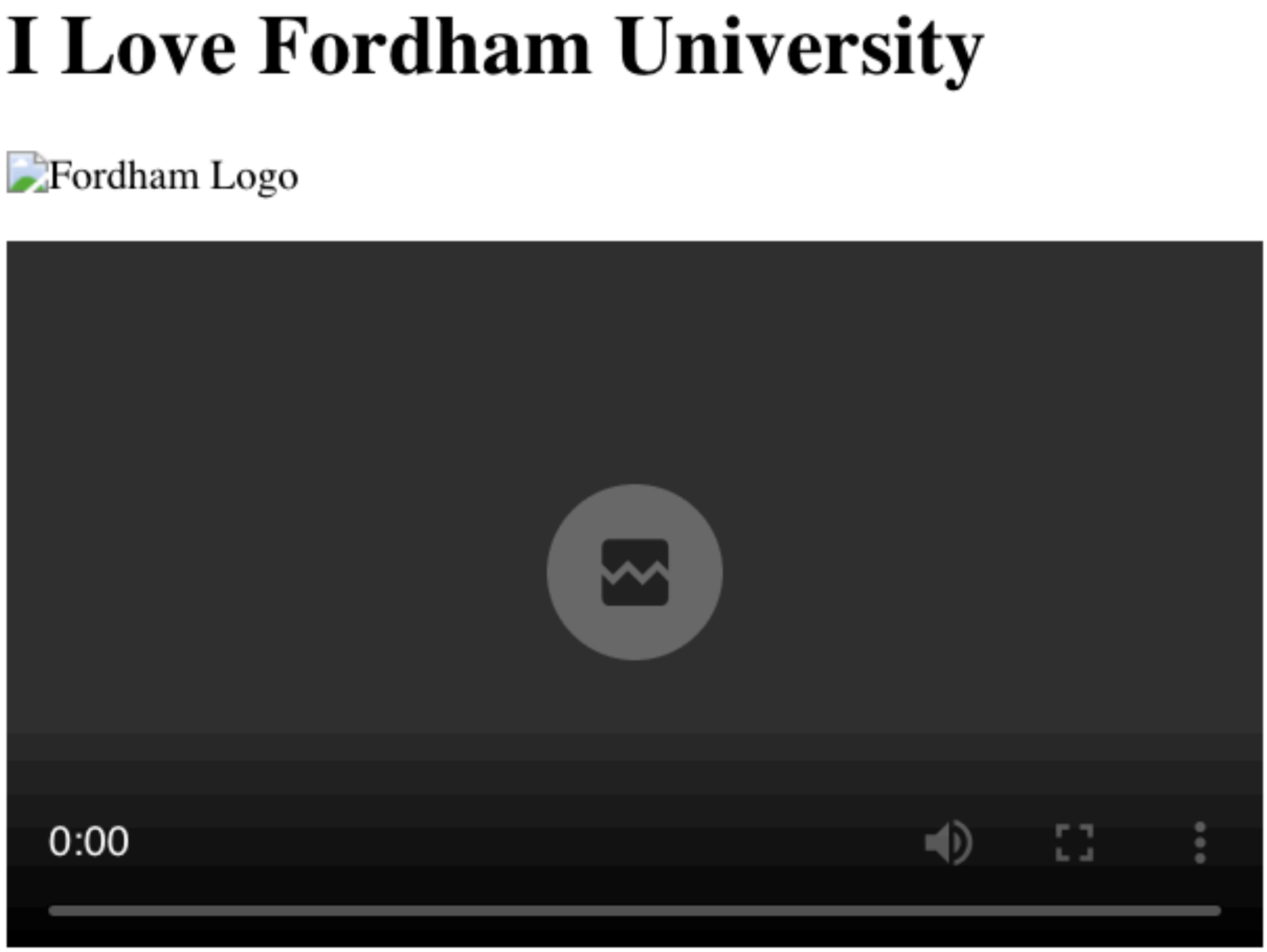
We care about the `status line` and one tag in the header, which is `content-type`. Depending on the type of file that you want to send, you can check the following link. For example, the type of css file should be `Content-Type: text/css` and the mp4 video should use `video/mp4` or `video/mpeg`

MIME types

Submission

- Your code for sender and receiver
- log file for sender and receiver.

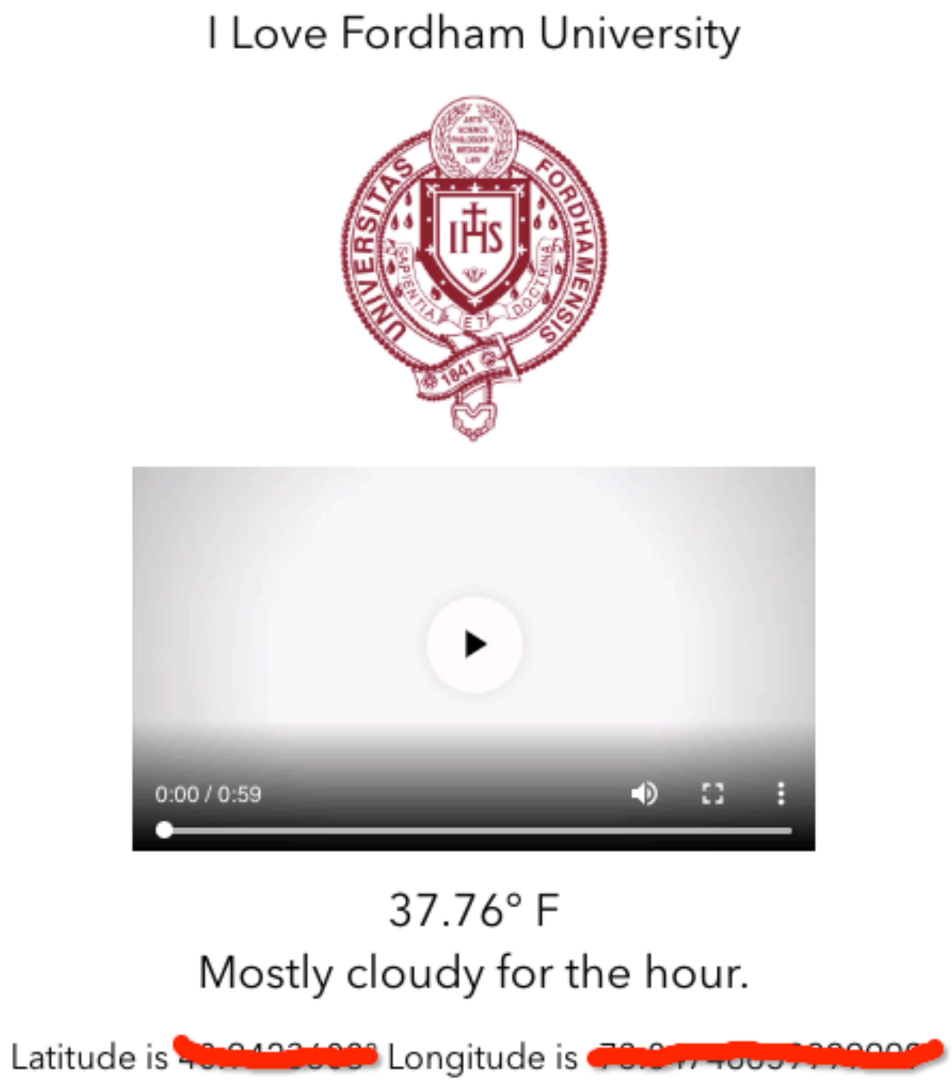
For example, if you just complete the first GET and send the index.html file back to browser, you will see the following webpage since the picture, video and js files are missing.



I Love Fordham University

Fordham Logo

If you complete all 4 GETs, you will see the following,



I Love Fordham University

37.76° F

Mostly cloudy for the hour.

Latitude is [redacted] Longitude is [redacted]

Submission

1. SimpleHttpServer.c
2. A detailed report
3. Demo on the last day of this semester (Demo is Due: 12.7 in class)

Grading Rubric

- 30% Get index.html (It's almost done with the sample code and the read file function)
- 20% Get Image
- 10% Get CSS file.
- 10% Get the video (may need a larger buffer).

- 10% Get JS.
- 10% Handle the status code and show client prop
- 10% The Report.

Since I design the project myself, you can use any useful information that you find online as long as you understand it and make it work.