COMP2322

Project Report

Cheung Kei Yau

23088621d

1. Design and Implementation

The multi-thread web server is implemented with python and tested on Windows. Python standard libraries `socket, threading, time, os, sys` and `datetime` is used.

The server handles HTTP requests from a client socket and open with the port. It supports serving text and image files from a specified directory `./web`. The server also handles persistent connections with 10 seconds timeout and supports the "If-Modified-Since" header for caching.

1.1 functions

➤ `available_port(host, port)` is used to find a port which is available recursively on the given host. If the initial port is used, the port will be added by 1 until there is an available port.

```
Running on http://127.0.0.1:8000
Socket is listening...
Connection from:  ('127.0.0.1', 54327)
```
[on Port 8000]

```
Running on http://127.0.0.1:8001
Socket is listening...
Connection from:  ('127.0.0.1', 57802)
```
[on Port 8001, Port 8000 is in use]

➤ `getHeader(status_code, file_type, last_modified):` generates HTTP response headers based on the status code, file type, and last modified time. There are a total of 6 response status: 200 OK, 304 Not Modified, 400 Bad Request, 403 Forbidden, 404 File Not Found and 415 Unsupported Media Type.

➤ `last_modified(file_path):` returns the last modified time of a file in HTTP date format.

➤ `webServer(client_socket, client_address):` handles client

requests, processes GET and HEAD methods, and sends appropriate HTTP responses.

1.2 hosting

The host and port number are set to 127.0.0.1 and 8000 respectively by default. The server listens for incoming connections on the specific host and port, handles each client connection in a separate thread, and serves the HTML and image files from the directory "web". It also records client hostname or IP address, access time, requested file name and response type for each record in the log file named "log.txt".
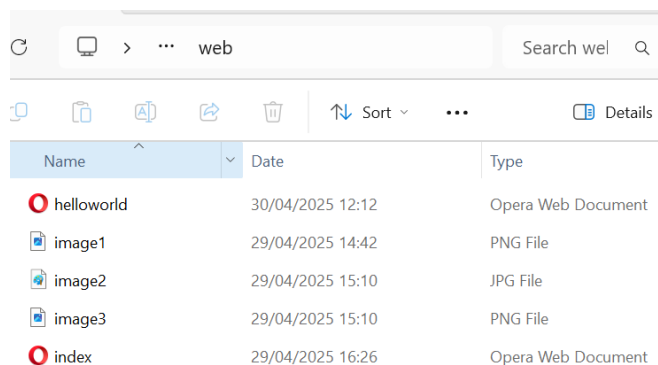
1.3 Keep-alive

The keep-alive timeout is set to 10 seconds and maximum100 seconds. The connection will be timeout if no request after 10 seconds. After a request, the web server will continue to receive requests from client and consider if it is persistent connection or not.

2. program execution
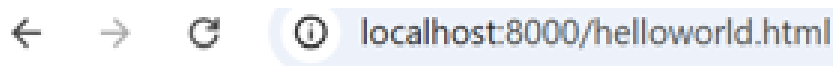
1. put the request source files into folder "web"

2. type "py source_code_23088621d.py" on terminal to start the server program.

```
PS C:\COMP2322\Project_23088621d> py source_code_23088621d.py
```

3. copy the website with port and visit this website on client. The URL should be "http://127.0.0.1:<PORT>/<request text or image file>". Below is an example of the URL:
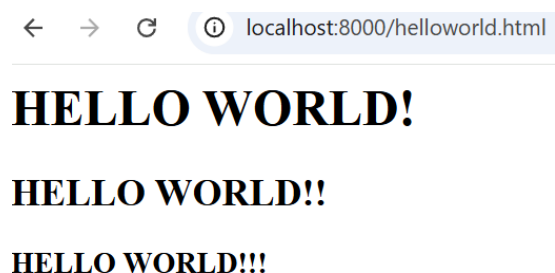
http://127.0.0.1:8000/helloworld.html

3. **Testing Cases**

To access the server, use a web browser, or an HTTP client such as curl or Postman. Make sure run the script before visiting the web.

Test for 200 ok


```
Method: GET
Request: /helloworld.html

Header:
HTTP/1.1 200 OK
Date: Wed, 30 Apr 2025 19:24:06
Connection: keep-alive
Keep-Alive: timeout=10, max=100
Last-Modified: Wed, 30 Apr 2025 12:12:02
Content-Type: text/html


Continue to receive requests.
Connection from:  ('127.0.0.1', 58210)
```



```
Method: GET
Request: /image1.png

Header:
HTTP/1.1 200 OK
Date: Wed, 30 Apr 2025 19:25:32
Connection: keep-alive
Keep-Alive: timeout=10, max=100
Last-Modified: Tue, 29 Apr 2025 14:42:05
Content-Type: image/png
```

Test for 404 File Not Found

找不到 localhost 網頁

找不到此網址的網頁：**http://localhost:8000/image2.png**

HTTP ERROR 404

```
Method: GET
Request: /image2.png

404 File Not Found
Header:
HTTP/1.1 404 Not Found
Date: Wed, 30 Apr 2025 19:26:01
Connection: keep-alive
Keep-Alive: timeout=10, max=100
Last-Modified: N/A
Content-Type: image/png
```

Test for 403 Forbidden

The test is done by changing the property of "image3", by denying read permission for authenticated users.
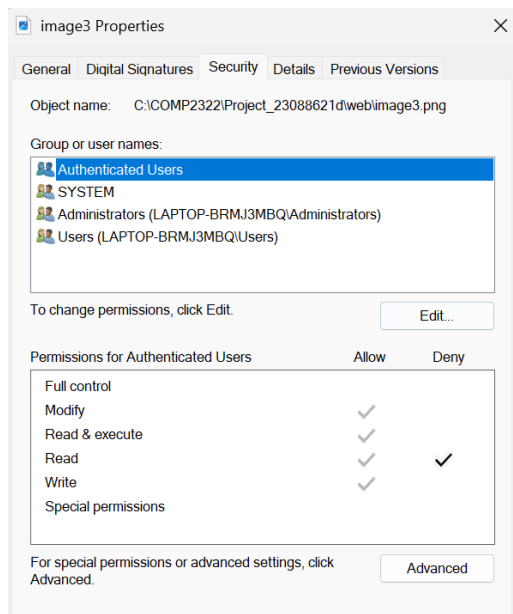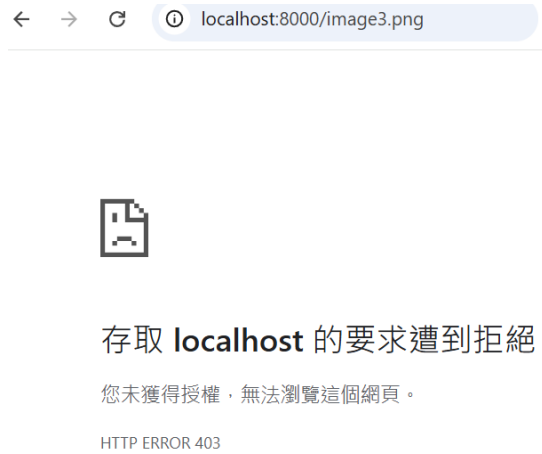
```
Method: GET
Request: /image3.png

403 Forbidden
Header:
HTTP/1.1 403 Forbidden
Date: Wed, 30 Apr 2025 19:27:06
Connection: keep-alive
Keep-Alive: timeout=10, max=100
Last-Modified: N/A
Content-Type: image/png
```

Test for 400 Bad Request


```
Method: GET
Request: /helloworld.html

400 Bad Request
Header:
HTTP/1.1 400 Bad Request
Date: Wed, 30 Apr 2025 13:01:29
Connection: keep-alive
Keep-Alive: timeout=10, max=100
Last-Modified: N/A
Content-Type: html
```

Test for 304 Not Modified

The test is done by "curl" command in the terminal.


```
C:\Users\cheun>curl -v -H "If-Modified-Since: Wed, 30 Apr 2025 12:00:00" http
://127.0.0.1:8000/index.html
*   Trying 127.0.0.1:8000...
* Connected to 127.0.0.1 (127.0.0.1) port 8000
* using HTTP/1.x
> GET /index.html HTTP/1.1
> Host: 127.0.0.1:8000
> User-Agent: curl/8.11.1
> Accept: */*
> If-Modified-Since: Wed, 30 Apr 2025 12:00:00
>
* Request completely sent off
< HTTP/1.1 304 Not Modified
< Date: Wed, 30 Apr 2025 13:17:16
< Connection: keep-alive
< Keep-Alive: timeout=10, max=100
< Last-Modified: Tue, 29 Apr 2025 16:26:16
< Content-Type: html
```

```
If-Modified-Since: Wed, 30 Apr 2025 12:00:00
Method: GET
Request: /index.html

304 Not Modified
Header:
HTTP/1.1 304 Not Modified
Date: Wed, 30 Apr 2025 13:17:16
Connection: keep-alive
Keep-Alive: timeout=10, max=100
Last-Modified: Tue, 29 Apr 2025 16:26:16
Content-Type: html
```

Test for 415 Unsupported Media Type

The test is done by requesting "image1.pdf" while pdf is not a supported type.

4. Log file

5. README file

```
 1    COMP2322 Multi-Threaded Web Server
 2    Cheung Kei Yau
 3    23088621d
 4
 5    #Usage
 6    1. put the request source files into folder - ./web
 7    2. type "py source_code_23088621d.py" on terminal to start the server program.
 8    3. copy the website with port and visit this website on client
 9
10    If the host already has a Web server running, use a port other than port 80.
11    The default port for the program is 8000.
12
13    #Features
14        - Multi-threaded Web server
15        - Proper request and response message exchanges
16        - GET command for both text files and image files
17        - HEAD command
18        - Six types of response statuses
19            - 200 OK
20            - 304 Not Modified
21            - 400 Bad Request
22            - 403 Forbidden
23            - 404 File Not Found
24            - 415 Unsupported Media Type
25        - Last-Modified and If-Modified-Since header fields
26        - Keep-Alive header field
```