

**Comp 2322**

**Computer Networking Project: Multi-thread Web Server**

**Due time: 11:59pm, April 30, 2025, Wednesday**

**Total marks: 100 marks**

**Student: Leung Chung Pang**

**Sid:23093488d**

## Summary

### 1. Architecture Overview

The multi-threaded HTTP server is built in Python using low-level socket programming. It follows a modular design with clear separation of concerns:

- **server.py**: Main entry point (handles socket setup, threading, and client connections).
- **http\_handler.py**: Processes HTTP requests/responses (parsing headers, generating responses).
- **file\_handler.py**: Manages filesystem operations (reading files, checking permissions).
- **utils.py**: Helper functions (logging, path resolution).

#### Key features:

- ✓ multi-threaded (handles concurrent clients)
- ✓ HTTP/1.1 compliant (supports GET, HEAD, status codes, headers)
- ✓ Persistent connections (Connection: keep-alive)
- ✓ Error handling (404, 403, 400, etc.)
- ✓ Request logging (client IP, timestamp, requested file, response code)

## **2. Core Components**

### **A. Socket & Threading (server.py)**

- Creates a TCP socket bound to 0.0.0.0:8080.
- Spawns a new thread for each client connection.
- Delegates request processing to http\_handler.py.

### **B. HTTP Protocol Logic (http\_handler.py)**

- Parses requests: Extracts method (GET/HEAD), path, and headers.
- Validates requests: Checks for malformed requests (400), unauthorized access (403).

#### **Generates responses:**

- 200 OK: Serves requested files (HTML, images, etc.).
- 304 Not Modified: Handles If-Modified-Since.
- 404/403: File not found or forbidden.

**Headers: Supports Content-Type, Last-Modified, Connection.**

### **C. Filesystem Operations (file\_handler.py)**

Checks file existence/permissions (returns 404/403 if invalid).

Reads files (binary mode for images, text for HTML).

Detects MIME types (e.g., image/png, text/html).

### **D. Utilities (utils.py)**

Logging: Records client IP, timestamp, and response codes to server.log.

Path resolution: Converts URL paths to filesystem paths safely.

Auto-creates www/: Ensures default index.html exists.

### 3. Key Technical Decisions

#### Multi-threading

Uses Python's **threading** for concurrent connections.

- Simple but effective for small-scale servers.

#### Low-Level Socket Programming

- Avoids high-level frameworks (e.g., Flask) to meet project requirements.
- Manually parses HTTP requests/responses.

#### Security

- Path sanitization: Prevents directory traversal attacks (e.g., blocking `/../`).
- Permission checks: Validates file readability before serving.

#### Connection Handling

- Supports both **close** (default) and **keep-alive** (reuses TCP connections).

## Screen capturing of results of all functions

### 1. Server Startup

```
23093488d-apollo:/home/23093488d/project$ python3 server.py
Server running on 127.0.0.1:8080
Serving files from: /home/23093488d/project/www
```

### 2. Basic GET Request (200 OK) {Test GET for Text File (index.html)}

```
23093488d-apollo:/home/23093488d$ curl -i http://127.0.0.1:8080/index.html
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 155
Last-Modified: Wed, 30 Apr 2025 12:34:40 GMT
Connection: close

<!DOCTYPE html>
<html>
<head><title>Python Web Server</title></head>
<body>
<h1>Welcome to Python Web Server</h1>
<p>Default index page</p>
</body>
</html>23093488d-apollo:/home/23093488d$
```

### 3. Test GET for Image File (test.jpeg):

```

</html>23093488d-apollo:/home/23093488d$ curl -i http://127.0.0.1:8080/test.jpeg
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 24162
Last-Modified: Wed, 30 Apr 2025 12:26:59 GMT
Connection: close

JFIF
ExifMM
$ &# " (-90 (*6+"#2D26;=@@&0FKE>J9?@=C

"=====

)!1AQa"q2#B#R$3br
%&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz

w!1AQa"2B#3R#br
$4%&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz

?Z (
(
(
(
(
(
(SU0<p_kj}
er}qQ:i" M*P,H!UE,
o|wahS=abC[sIkMw=O<
@Fs3tG=-l2sqa=v=wd=Fa=b6=f9.v=z9=w=KN(c=CWhH\,

```

#### 4. Test 415 for Unsupported File (test.exe):

```

23093488d-apollo:/home/23093488d$ curl -i http://127.0.0.1:8080/test.exe
HTTP/1.1 415 Unsupported Media Type
Connection: close

```

#### 5. Test HEAD Request:

```

23093488d-apollo:/home/23093488d$ curl -I http://127.0.0.1:8080/index.html
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 155
Last-Modified: Wed, 30 Apr 2025 12:34:40 GMT
Connection: close

```

```

23093488d-apollo:/home/23093488d$ █

```

#### 6. Test 404 for Missing File:

```

23093488d-apollo:/home/23093488d$ curl -i http://127.0.0.1:8080/missing.txt
HTTP/1.1 404 File Not Found
Connection: close

```

#### 7. Test 304 Not Modified (If-Modified-Since):

```
23093488d-apollo:/home/23093488d$ curl -I http://127.0.0.1:8080/index.html
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 155
Last-Modified: Wed, 30 Apr 2025 12:34:40 GMT
Connection: close

23093488d-apollo:/home/23093488d$ curl -i -H "If-Modified-Since: Wed, 30 Apr 2025 12:34:56 GMT" http://127.0.0.1:8080/index.html
HTTP/1.1 304 Not Modified
Connection: close
```

## 8. Test 400 for Bad Request

```
23093488d-apollo:/home/23093488d$ which nc
/bin/nc
23093488d-apollo:/home/23093488d$ echo -e "GET\r\n\r\n" | nc 127.0.0.1 8080
HTTP/1.1 400 Bad Request
Connection: close
```

## 9. Test 403 for forbidden request

Since I am using apollo so I cannot change the permission of the owner so I can't capture the progress. But it should work on your computer.

## 10. Persistent Connection (keep-alive, 5 marks):

```
23093488d-apollo:/home/23093488d$ curl -H "Connection: keep-alive" http://127.0.0.1:8080/index.html
<!DOCTYPE html>
<html>
<head><title>Python Web Server</title></head>
<body>
<h1>Welcome to Python Web Server</h1>
<p>Default index page</p>
</body>
</html>23093488d-apollo:/home/23093488d$ curl -H "Connection: keep-alive" http://127.0.0.1:8080/index.html
<!DOCTYPE html>
<html>
<head><title>Python Web Server</title></head>
<body>
<h1>Welcome to Python Web Server</h1>
<p>Default index page</p>
</body>
</html>23093488d-apollo:/home/23093488d$ curl -H "Connection: keep-alive" http://127.0.0.1:8080/index.html http://127.0.0.1:8080/index.html -o /dev/null -s -w "%{http_code}\n"
200
<!DOCTYPE html>
<html>
<head><title>Python Web Server</title></head>
<body>
<h1>Welcome to Python Web Server</h1>
<p>Default index page</p>
</body>
</html>200
23093488d-apollo:/home/23093488d$ netstat -an | grep 8080
tcp        0      0 127.0.0.1:8080      0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:40256     127.0.0.1:8080       TIME_WAIT
tcp        0      0 127.0.0.1:40250     127.0.0.1:8080       TIME_WAIT
tcp        0      0 127.0.0.1:40252     127.0.0.1:8080       TIME_WAIT
```

```
23093488d-apollo:/home/23093488d$ curl -v -H "Connection: keep-alive" http://127.0.0.1:8080/index.html
* About to connect() to 127.0.0.1 port 8080 (#0)
*   Trying 127.0.0.1...
* Connected to 127.0.0.1 (127.0.0.1) port 8080 (#0)
> GET /index.html HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 127.0.0.1:8080
> Accept: */*
> Connection: keep-alive
>
< HTTP/1.1 200 OK
< Content-Type: text/html
< Content-Length: 155
< Last-Modified: Wed, 30 Apr 2025 12:34:40 GMT
< Connection: keep-alive
<
<!DOCTYPE html>
<html>
<head><title>Python Web Server</title></head>
<body>
<h1>Welcome to Python Web Server</h1>
<p>Default index page</p>
</body>
* Connection #0 to host 127.0.0.1 left intact
```

```
23093488d-apollo:/home/23093488d$ cat project/server.log
2025-04-30 20:39:37 - 127.0.0.1 - /index.html - 200
2025-04-30 20:40:39 - 127.0.0.1 - /index.html - 200
2025-04-30 20:41:26 - 127.0.0.1 - /index.html - 200
2025-04-30 20:48:08 - 127.0.0.1 - /test.jpeg - 200
2025-04-30 20:48:57 - 127.0.0.1 - /test.jpeg - 200
2025-04-30 20:49:31 - 127.0.0.1 - /test.jpeg - 200
2025-04-30 20:57:52 - 127.0.0.1 - /index.html - 200
2025-04-30 20:58:16 - 127.0.0.1 - /test.exe - 415
2025-04-30 13:03:14 - 127.0.0.1 - /test.exe - 415
2025-04-30 13:03:59 - 127.0.0.1 - /index.html - 200
2025-04-30 13:06:22 - 127.0.0.1 - /missing.txt - 404
2025-04-30 13:07:10 - 127.0.0.1 - /index.html - 200
2025-04-30 13:09:22 - 127.0.0.1 - /index.html - 200
2025-04-30 13:11:24 - 127.0.0.1 - /index.html - 200
2025-04-30 13:15:03 - 127.0.0.1 - /index.html - 200
2025-04-30 13:22:05 - 127.0.0.1 - /index.html - 200
2025-04-30 13:24:46 - 127.0.0.1 - /index.html - 304
2025-04-30 13:26:24 - 127.0.0.1 - /index.html - 304
2025-04-30 13:27:50 - 127.0.0.1 - /index.html - 200
2025-04-30 13:28:07 - 127.0.0.1 - /index.html - 304
2025-04-30 13:33:54 - 127.0.0.1 - /index.html - 304
2025-04-30 13:37:25 - 127.0.0.1 - /restricted.txt - 200
2025-04-30 13:55:17 - 127.0.0.1 - /restricted.txt - 200
2025-04-30 14:12:46 - 127.0.0.1 - /index.html - 200
2025-04-30 14:14:47 - 127.0.0.1 - /test.jpeg - 200
2025-04-30 14:16:25 - 127.0.0.1 - /test.exe - 415
2025-04-30 14:17:34 - 127.0.0.1 - /index.html - 200
2025-04-30 14:18:20 - 127.0.0.1 - /missing.txt - 404
2025-04-30 14:18:59 - 127.0.0.1 - /index.html - 200
2025-04-30 14:19:03 - 127.0.0.1 - /index.html - 304
2025-04-30 14:40:34 - 127.0.0.1 - /restricted.txt - 200
2025-04-30 15:24:30 - 127.0.0.1 - /index.html - 200
2025-04-30 15:24:36 - 127.0.0.1 - /index.html - 200
2025-04-30 15:31:07 - 127.0.0.1 - /index.html - 200
2025-04-30 15:31:12 - 127.0.0.1 - /index.html - 200
2025-04-30 15:31:22 - 127.0.0.1 - /index.html - 200
2025-04-30 15:31:22 - 127.0.0.1 - /index.html - 200
2025-04-30 15:32:22 - 127.0.0.1 - /index.html - 200
```



## 11. Non-Persistent Connection (close, 5 marks):

```
23093488d-apollo:/home/23093488d$ curl -v -H "Connection: close" http://127.0.0.1:8080/index.html
* About to connect() to 127.0.0.1 port 8080 (#0)
*   Trying 127.0.0.1...
* Connected to 127.0.0.1 (127.0.0.1) port 8080 (#0)
> GET /index.html HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 127.0.0.1:8080
> Accept: */*
> Connection: close
>
< HTTP/1.1 200 OK
< Content-Type: text/html
< Content-Length: 155
< Last-Modified: Wed, 30 Apr 2025 12:34:40 GMT
< Connection: close
<
<!DOCTYPE html>
<html>
<head><title>Python Web Server</title></head>
<body>
<h1>Welcome to Python Web Server</h1>
<p>Default index page</p>
</body>
* Closing connection 0
```

```
</html>23093488d-apollo:/home/23093488d$ netstat -an | grep 8080
tcp        0      0 127.0.0.1:8080      0.0.0.0:*            LISTEN
tcp        0      0 127.0.0.1:40290     127.0.0.1:8080       TIME_WAIT
23093488d-apollo:/home/23093488d$ cat project/server.log
2025-04-30 20:39:37 - 127.0.0.1 - /index.html - 200
2025-04-30 20:40:39 - 127.0.0.1 - /index.html - 200
2025-04-30 20:41:26 - 127.0.0.1 - /index.html - 200
2025-04-30 20:48:08 - 127.0.0.1 - /test.jpeg - 200
2025-04-30 20:48:57 - 127.0.0.1 - /test.jpeg - 200
2025-04-30 20:49:31 - 127.0.0.1 - /test.jpeg - 200
2025-04-30 20:57:52 - 127.0.0.1 - /index.html - 200
2025-04-30 20:58:16 - 127.0.0.1 - /test.exe - 415
2025-04-30 13:03:14 - 127.0.0.1 - /test.exe - 415
2025-04-30 13:03:59 - 127.0.0.1 - /index.html - 200
2025-04-30 13:06:22 - 127.0.0.1 - /missing.txt - 404
2025-04-30 13:07:10 - 127.0.0.1 - /index.html - 200
2025-04-30 13:09:22 - 127.0.0.1 - /index.html - 200
2025-04-30 13:11:24 - 127.0.0.1 - /index.html - 200
2025-04-30 13:15:03 - 127.0.0.1 - /index.html - 200
2025-04-30 13:22:05 - 127.0.0.1 - /index.html - 200
2025-04-30 13:24:46 - 127.0.0.1 - /index.html - 304
2025-04-30 13:26:24 - 127.0.0.1 - /index.html - 304
2025-04-30 13:27:50 - 127.0.0.1 - /index.html - 200
2025-04-30 13:28:07 - 127.0.0.1 - /index.html - 304
2025-04-30 13:33:54 - 127.0.0.1 - /index.html - 304
2025-04-30 13:37:25 - 127.0.0.1 - /restricted.txt - 200
2025-04-30 13:55:17 - 127.0.0.1 - /restricted.txt - 200
2025-04-30 14:12:46 - 127.0.0.1 - /index.html - 200
2025-04-30 14:14:47 - 127.0.0.1 - /test.jpeg - 200
2025-04-30 14:16:25 - 127.0.0.1 - /test.exe - 415
2025-04-30 14:17:34 - 127.0.0.1 - /index.html - 200
2025-04-30 14:18:20 - 127.0.0.1 - /missing.txt - 404
2025-04-30 14:18:59 - 127.0.0.1 - /index.html - 200
2025-04-30 14:19:03 - 127.0.0.1 - /index.html - 304
2025-04-30 14:40:34 - 127.0.0.1 - /restricted.txt - 200
2025-04-30 15:24:30 - 127.0.0.1 - /index.html - 200
2025-04-30 15:24:36 - 127.0.0.1 - /index.html - 200
2025-04-30 15:31:07 - 127.0.0.1 - /index.html - 200
2025-04-30 15:31:12 - 127.0.0.1 - /index.html - 200
2025-04-30 15:31:22 - 127.0.0.1 - /index.html - 200
2025-04-30 15:31:22 - 127.0.0.1 - /index.html - 200
2025-04-30 15:32:22 - 127.0.0.1 - /index.html - 200
2025-04-30 15:36:32 - 127.0.0.1 - /index.html - 200
```