

Assignment 2: Build a Meme-Generating Proxy Server (10%)

Start Date: 14th February 2025

Submission Deadline: 28th February 2025 (11.59 PM)

Submission Medium: GradeScope

Background

Imagine Mochi the panda 🐼 browsing his favorite bamboo recipe website, only to discover every image magically replaced with hilarious memes or pictures of his funny panda friends! In this assignment, you'll help Mochi build a proxy server that intercepts HTTP (NOT HTTPS!) responses and swaps some or all of the images with memes or creative surprises. Mochi is counting on you to use your networking skills and a touch of humor to make his browsing experience unforgettable!

Objective

The goal of this assignment is to create a Meme-generating proxy server that not only forwards HTTP requests but also injects fun memes into websites. This is a playful way to learn about HTTP, response modification, and web data manipulation.

Assignment Tasks

- Setup and Initialization
 - Set up a proxy server that listens for client connections and forwards HTTP requests.
 - Handle multiple client connections using threads or asynchronous programming.
- Image Interception
 - Identify image requests (e.g., `.jpg`, `.png`, `.gif`, etc`) in HTTP responses.
 - Replace 50% of the images on a webpage (e.g.: <http://httpbin.org/>) with memes from a pre-defined folder or URL list.
- Meme Pool
 - Create a folder containing at least 15 memes to use as replacements.
 - Randomly select memes from the pool to inject into the webpages.
- Personalized Easter Egg
 - Add a special Easter egg feature: when a specific URL (e.g., <http://google.ca>) is requested, return a custom surprise page with a full-screen meme, sound effects, or fun messages.
- Error Handling
 - Ensure that the proxy doesn't crash when it encounters unsupported file types, invalid requests, or server errors (such as <https://.....>).

Technical Requirements

- **Networking:** Use TCP sockets for communication. Ensure the server listens on a configurable port.
- **Image Handling:** Analyze HTTP responses and detect image content types (e.g., JPEG, PNG, webp, etc). Replace image URLs or embed Base64-encoded meme images directly into the HTML response.
- **Concurrency:** Handle multiple client connections simultaneously.
- **Testing Memes:** Make sure that memes appear in place of original images on websites (<http://httpbin.org/>) when viewed through the proxy.

Evaluation Criteria (Grading Rubric)

The assignment will be graded out of 30 points.

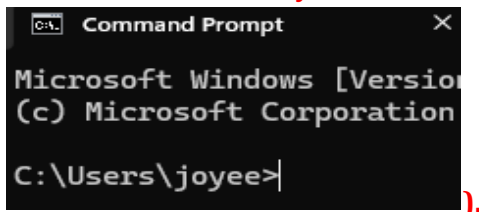
- **Functionality (15 pts):**
 - **Proxy Server Operation (3 pts):** The proxy listens on the correct address (e.g., 127.0.0.1:8080) and forwards HTTP requests to the correct destination.
 - **Meme Replacement (6 pts):** The proxy server intercepts image requests and replaces them with random memes from the memes folder.
 - **Easter Egg (6 pts):** When the server detects a request for google.ca, it should respond with a meme image.
- **Code Quality (6 pts):**
 - **Code Structure and Organization (2 pts):** The code should be cleanly separated into functions, with logical organization. For instance, there should be separate functions for handling client requests, generating memes, etc.
 - **Variable Naming and Commenting (2 pts):** Variables should have clear and descriptive names (e.g., client_socket, meme_folder). Proper comments are expected in areas of complex logic or important functions.
 - **Error Handling and Robustness (2 pts):** The code should handle common edge cases, such as missing Host headers, empty meme folders, malformed requests, etc.
- **Documentation and Demo (9 pts):**
 - **README and Setup Instructions (4 pts):** The README file should explain how to set up the project, configure the browser for testing, and run the server. It should also include any additional information, such as prerequisites or dependencies.
 - **Demo (5 pts):** A video demo of the proxy server code in action.

Testing Details

- Basic Functionality: Use a Firefox browser to visit <http://httpbin.org/> through the proxy. Check that original images are partially replaced by memes.
- Easter Egg: Visit <http://google.ca> and verify that the surprise page loads correctly.
- Error Scenarios: Test with unsupported website <https://www.google.ca/> and ensure the proxy handles them without crashing.
- Creativity: Check the diversity of memes and creativity of the Easter egg.

Submission Requirements

- The Python source code with comments explaining the logic in Gradescope.
- A `README` file explaining how to run and test the program in Gradescope.
- A folder containing your meme pool (minimum 15 images) in Gradescope.
- A video demo of the proxy in action in Gradescope. **(Your code should be executed in the command prompt, ensuring that your computer name is displayed as your identifier to verify that the work is genuinely yours:**



```
Microsoft Windows [Version 10.0.17134.0]
(c) Microsoft Corporation
C:\Users\joyee>
```