Programming Assignment 2: Web Server

Due: 03/28/2025 11:59 PM

Objectives

In this assignment, you will learn the basics of socket programming for TCP connections in Python: how to create a socket, bind it to a specific address and port, as well as send and receive an HTTP packet. You will also learn some basics of HTTP header format.

Detailed instructions

The main task is to write a command line tool that accepts simple HTTP requests and responds by retrieving the requested object from a given local directory. The tool will function as a simple HTTP server that can serve HTTP client connections. Specifically, the server is expected to

(1) Use server-side TCP sockets (do not use high-level libraries like urllib or similar to accept client connections).
(2) Receive and correctly respond to valid HTTP GET requests.
(3) Send an HTTP "404 Not Found" message back to the client, if the requested file is not present in the server.

Here is the set of command line options that must be supported:

```
usage: webserver.py [-h] -r OBJECT_DIR [-p PORT]

Start a web server

optional arguments:
  -h, --help      show this help message and exit
  -r OBJECT_DIR  Root directory for files
  -p PORT         The port number on which the server will be listening for
                  incoming connections
```

Example Usage:

```
./webserver.py -r object_dir -p server_port
```

The objects_dir directory will be provided during testing, and should be used as the root directory for the URLs provided by the client in the GET requests.

For instance, say that object_dir looks like the following:

```
-- object_dir
       |_ file_1.html
       |_ file_2.png
       |_ ...
       |_ subdir_1
       |___ subfile_1.jpg
       |_ subdir_2
       |___ subfile_2.html
       |_ ...
```

A request for http://<server_ip>:<server_port>/subdir_1/subfile_1.jpg must return the content of the object_dir/subdir_1/subfile_1.jpg. if the requested file is not present in the server, the server should send an HTTP “404 Not Found” message back to the client.

Submission

Create a directory named with your last name. Put your program's files under that directory. Then, create a .zip file containing that directory. Finally, submit the zip file through Canvas under this assignment.

Late policy for this assignment

Less than 24 hours late -- 20% penalty. 24 to 48 hours late -- 40% penalty. Later than 48 hours -- 0 pt.