EE129 Project 2: A Simple Web Server in Python

Notes 7 (10/28/24)

1 Introduction

Python is a popular, general-purpose programming language that is easy to learn and program. Web technology is a key technology that revolutionized the Internet in the 1990s and has been fueling the development of the Internet for more than 20 years. A web server and a client (web browser) communicate with a TCP-based protocol called HTTP (Hypertext Transfer Protocol). We will use Python to write a simple web server from scratch!

2 Before starting your program

1. Download Python. The `Anaconda` edition is recommended. This edition supports Windows, Mac OSX, and Linux operating systems and includes all the packages you need, and more packages for your future Python programming. Choose the right file for your computer. Please use Python 3.X edition as we will work in Python 3. Don't use Python 2 edition. Python 2 and Python 3 are not completely compatible.
2. Study the Python documentation or tutorials for socket programming in Python. Socket is the API (Application Programming Interface) for various network protocols, including the TCP/IP protocol suite.
3. Study the HTTP protocol by reading Section 2.2 of your textbook.
4. Study the HTML tutorials to learn the basics of writing a web page. Pay attention to HTML forms. We will use an HTML form to send data from the browser to the web server.

```
# Import socket module

from socket import *

import sys # In order to terminate the program


# Create a TCP server socket

#(AF_INET is used for IPv4 protocols)

#(SOCK_STREAM is used for TCP)


serverSocket = socket(AF_INET, SOCK_STREAM)


# Assign a port number
```

```
serverPort = 6789

# Bind the socket to server address and server port
serverSocket.bind(("", serverPort))

# Listen to at most 1 connection at a time
serverSocket.listen(1)

# Server should be up and running and listening to the incoming connections

while True:
    print('The server is ready to receive')

    # Set up a new connection from the client
    connectionSocket, addr = serverSocket.accept()

    # If an exception occurs during the execution of try clause
    # the rest of the clause is skipped
    # If the exception type matches the word after except
    # the except clause is executed
    try:
      # Receives the request message from the client
      message = connectionSocket.recv(1024).decode()
      # Extract the path of the requested object from the message
      # The path is the second part of HTTP header, identified by [1]
      filename = message.split()[1]
      # Because the extracted path of the HTTP request includes
      # a character '\', we read the path from the second character
      f = open(filename[1:])
      # Store the entire contenet of the requested file in a temporary buffer
      outputdata = f.read()
```

```
            # Send the HTTP response header line to the connection socket
            connectionSocket.send("HTTP/1.1 200 OK\r\n\r\n".encode())

            # Send the content of the requested file to the connection socket
            for i in range(0, len(outputdata)):
                    connectionSocket.send(outputdata[i].encode())
            connectionSocket.send("\r\n".encode())

            # Close the client connection socket
            connectionSocket.close()

        except IOError:
                # Send HTTP response message for file not found
                connectionSocket.send("HTTP/1.1 404 Not Found\r\n\r\n".encode())
                connectionSocket.send("<html><head></head><body><h1>404 Not
Found</h1></body></html>\r\n".encode())
                # Close the client connection socket
                connectionSocket.close()

    serverSocket.close()
    sys.exit()#Terminate the program after sending the corresponding data
```

3 Project details

A. Study and run the web server program giving in class. Modify the program to send an image file to the client.

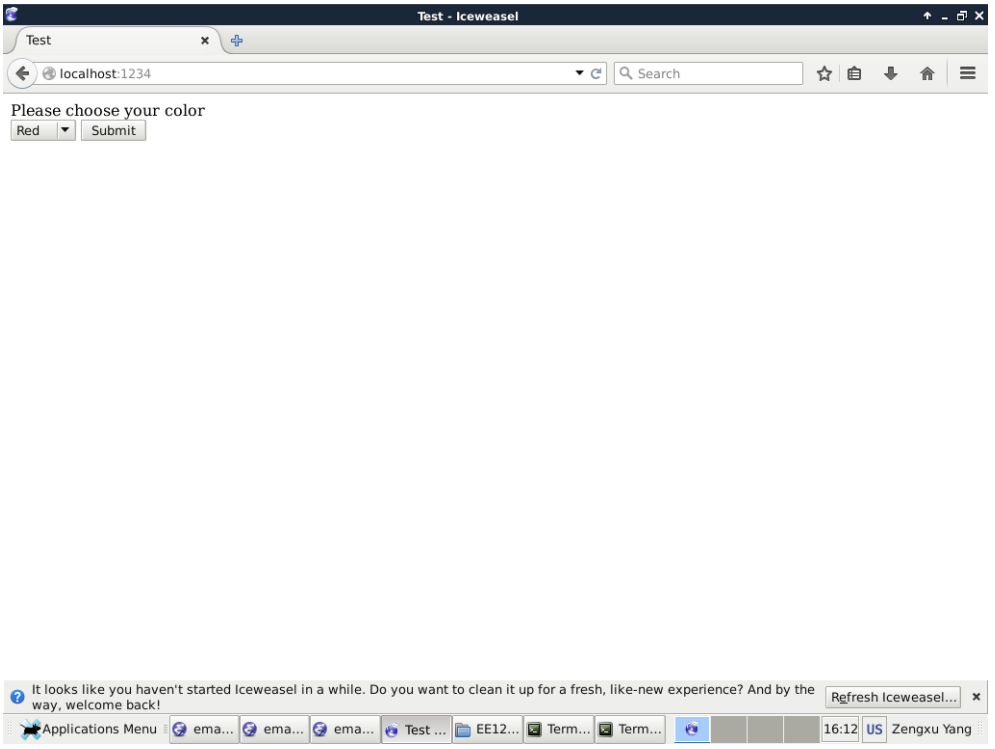
B. Another web server.

1. Write a simple web server using sockets in Python that can communicate with a web browser in HTTP 1.1 protocol. When specifying the TCP port, use a port number in the range of 1024-65535 so that no special privilege is needed for your server. No other Python library is allowed except the `socket` library. Add the following line at the beginning of your code to use the `socket` library:

```
import socket
```

2. When requested by a web browser, your web server should check the URL sent by your browser and act accordingly:

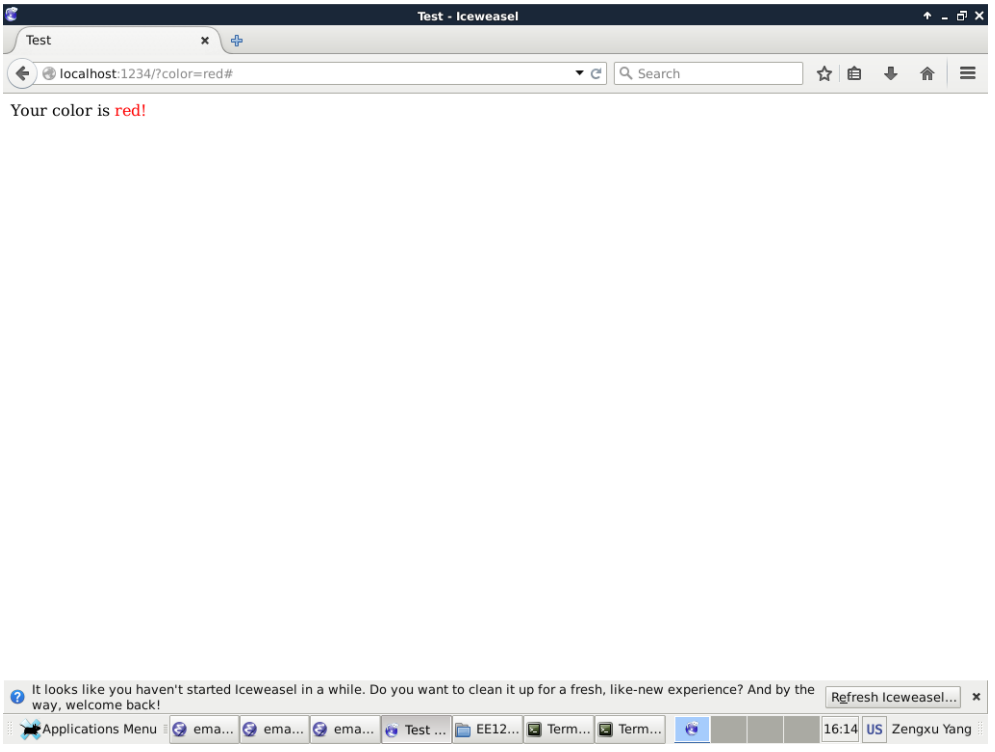
   (a) If the URL is just "/", it should display a page with a form for the user to select from 2 different colors (red, green), then submit it to your web server with the GET method.



   (b) If the URL has the keyword "red", it should send back a web page with a sentence:

```
Your color is red!
```

   with the red color.

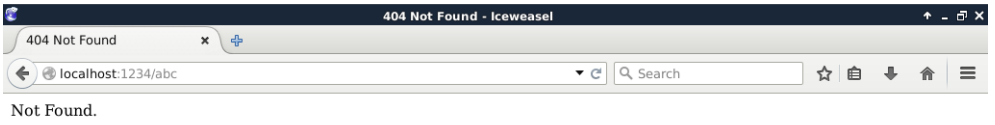


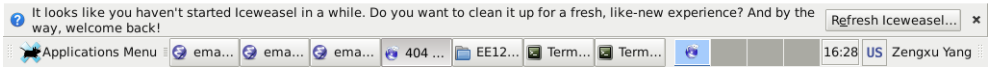
(c) If the URL has the keyword "green", it should send back a web page with a sentence:

```
Your color is green!
```

with the green color.

3. Otherwise, the user requested a page that does not exist. Return the status code 404 to the web browser.





4 Tips and extra credits

1. The total length of your program should be less than 100 line. If your program is too long, it is overly complicated. Rethink and cut the length.
2. Run your program by typing `python filename` or `python3 filename` depending on your installation method. Or for some platforms, you can just double click your source file to run it.
3. If your progrom fails because the specified port has been used by other programs, change your port number in your code and run it again.
4. You can visit your web server from a browser on the same computer by using the special domain name `localhost`. Or you can visit your web server from another computer on the same LAN by using your computer's IP address.
5. Use your imagination to add more features to the project to get extar crdeit. Descirbe them in your report.

5 Submission

Show the working of your program to TA before November 13th during her office hours and submit a report on introduction, how it works, the results and discussion with a well-commented program by 11:59 pm, Friday, November 15th, 2024. A report format guideline is on Canvas.

Enjoy your project. Have fun!