

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

تمرین سری ششم: برنامه نویسی سوکت

تمرین انتهای فصل دوم کراس

امین خانی

aminkhai2010@gmail.com

شبکه های کامپیوتری، استاد گلی

دانشگاه کاشان

اردیبهشت ۱۴۰۲

صورت سوال

Assignment 1: Web Server

In this assignment, you will develop a simple Web server in Python that is capable of processing only one request. Specifically, your Web server will (i) create a connection socket when contacted by a client (browser); (ii) receive the HTTP request from this connection; (iii) parse the request to determine the specific file being requested; (iv) get the requested file from the server's file system; (v) create an HTTP response message consisting of the requested file preceded by header lines; and (vi) send the response over the TCP connection to the requesting browser. If a browser requests a file that is not present in your server, your server should return a "404 Not Found" error message.

In the companion Web site, we provide the skeleton code for your server. Your job is to complete the code, run your server, and then test your server by sending requests from browsers running on different hosts. If you run your server on a host that already has a Web server running on it, then you should use a different port than port 80 for your Web server.

در این تکلیف، شما یک وب سرور ساده در پایتون ایجاد خواهید کرد که قادر به پردازش تنها یک درخواست است. به طور خاص، سرور وب شما (i) هنگامی که مشتری (مرورگر) با آن تماس می گیرد، یک سوکت اتصال ایجاد می کند. (ii) درخواست HTTP را از این اتصال دریافت کنید. (iii) درخواست را برای تعیین پرونده خاص مورد درخواست تجزیه و تحلیل کند. (iv) دریافت فایل درخواستی از سیستم فایل سرور؛ (v) ایجاد یک پیام پاسخ HTTP متشکل از فایل درخواستی که قبل از خطوط سرصفحه قرار دارد. و (vi) پاسخ را از طریق اتصال TCP به مرورگر درخواست کننده ارسال کنید. اگر مرورگر فایلی را درخواست کند که در سرور شما وجود ندارد، سرور شما باید پیام خطای "404 یافت نشد" را برگرداند.

در وب سایت همراه، ما کد اسکلت سرور شما را ارائه می دهیم. وظیفه شما این است که کد را تکمیل کنید، سرور خود را اجرا کنید و سپس سرور خود را با ارسال درخواست از مرورگرهای در حال اجرا بر روی هاست های مختلف آزمایش کنید. اگر سرور خود را روی میزبانی اجرا می کنید که قبلاً یک وب سرور روی آن در حال اجرا است، باید از پورت متفاوتی نسبت به پورت 80 برای وب سرور خود استفاده کنید.

server.py

```
1 #import socket module
2 from socket import *
3 serverSocket = socket(AF_INET, SOCK_STREAM)
4 #Prepare a sever socket
5 serverPort = 80
6 serverSocket.bind("", serverPort)
7 serverSocket.listen(1)
8 while True:
9     #Establish the connection
10    print('Ready to serve...')
11    connectionSocket, addr = serverSocket.accept()
12    try:
13        message = connectionSocket.recv(1024)
14        filename = message.split()[1]
15        f = open(filename[1:])
16        outputdata = f.read()
17        f.close()
18        #Send one HTTP header line into socket
19        outputdata = 'HTTP/1.1 200 OK\r\n\r\n' + outputdata
20        #Send the content of the requested file to the client
21        for i in range(0, len(outputdata)):
22            connectionSocket.send(outputdata[i].encode())
23        connectionSocket.close()
24        print("OK!")
25    except IOError:
26        #Send response message for file not found
27        outputdata = 'HTTP/1.1 404 Not Found\r\n\r\n'
28        #Close client socket
29        for i in range(0, len(outputdata)):
30            connectionSocket.send(outputdata[i].encode())
31        connectionSocket.close()
32 serverSocket.close()
```

client.py

```
1 from socket import *
2
3 serverName = '127.0.0.1'
4 serverPort = 80
5 clientSocket = socket(AF_INET, SOCK_STREAM)
6 clientSocket.connect((serverName, serverPort))
7 outputdata = 'GET /hello.html HTTP/1.1\r\nHost: 127.0.0.1\r\n\r\n'
8 clientSocket.send(outputdata.encode())
9 data = 1
10 while data:
11     data = clientSocket.recv(1024)
12     print(data.decode(), end = '')
13 clientSocket.close()
```