Computer Network Homework2

20161190 이민재

1. Introduction / Enviroment

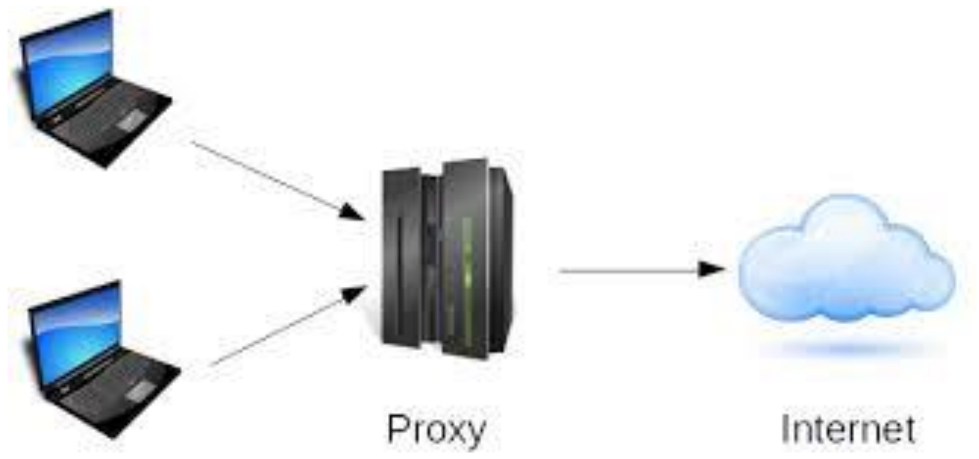
My homework is implementing an HTTP proxy. I performed using python source code and telnet(clinet). My execution environment is as follows.

- Source code : Python 2.7.16
- OS : MacBook Pro(13-inch, 2017, Two thunderbolt 3 ports)
- Processor : 2.3GHz dual core intel Core i5
- Memmory : 8GB 2133 MHZ LPDDR3
- Graphic : Intel Iris Plus Graphics 640 1536 MB
- Library : socket, thread, re, requests, ssl
-

If who can't implement import requests. Then using command "pip install requests" for install module.

2. Concep of proxy server

Proxy server is server that forward client request to the remote server and return response to the client. Proxy server is used for security reasons and efficiency reasons. Security reason is to prevent client from directly accessing to origin server. In terms of efficiency, Proxy server has the information that client frequently requests. Then if client request information that has already proxy server, proxy server doesn't request to origin server and directly respone to client.



-

3. Explanation of functions

Before explaining the source code in detail, I summerize representative socket-related functions.

1) socket(socket.AF_INET, socket.SOCK_STREAM)

This function can be used to create socket objects. In order to perform networking using sockets in the same way as servers or clients, it is necessary to create sockets first. This function receives two factors, one family and the other type. Socket.AF_INET is intended to represent an IPv4 address scheme, and socket_SOCK_STREAM means creating a socket using TCP.

2) setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

Sockets can adjust details by manipulating socket options, and for this purpose, a setsockopt() function can be used. When a server program using a socket is operated, there are cases where it is forcibly terminated or abnormally terminated. For testing purposes, it is especially often necessary to force termination, and if the program is forced to end and executed again, the existing program was terminated, but the kernel still maintains bind information with abnormal termination. This may often be associated with timeout in a TCP connection, which usually disappears after a minute, but is inserted to prevent connection delays during that time.

3) bind("127.0.0.7", 8080)

In general, server sockets use fixed port numbers. And the client's connection request is accepted with that port number. Therefore, the operating system must bind the socket and the port number to make the server socket use a specific port number, and the API used at this time is bind(). Currently, "127.0.0.7" means that the IP address of the local host is 127.0.0.7 and 8080 means the port number.

4) listen(5)

When bind() is finished, it is a function for making the client wait for connection. The listen() function receives the number of clients to be requested for connection as a parameter. In this case, it means that it will allow connection to five clients.

5) connect(host, port)

The client uses socket.connect() to connect to the server, and the factor used at this time is the same as bind(). This method ends when a connection is established, and there is no value to return. Since the client must be connected to the server, receive the local host address through sys and enter the same port number previously created on the server side.

6) accept()

This function is used for connection-oriented sockets. It takes the first connection from the queue where the connections that have not yet been processed are waiting and creates a new connection socket. And assign a file designator pointing to the socket and return it. In this case, the information on the socket, the IP address, and the Port number are returned.

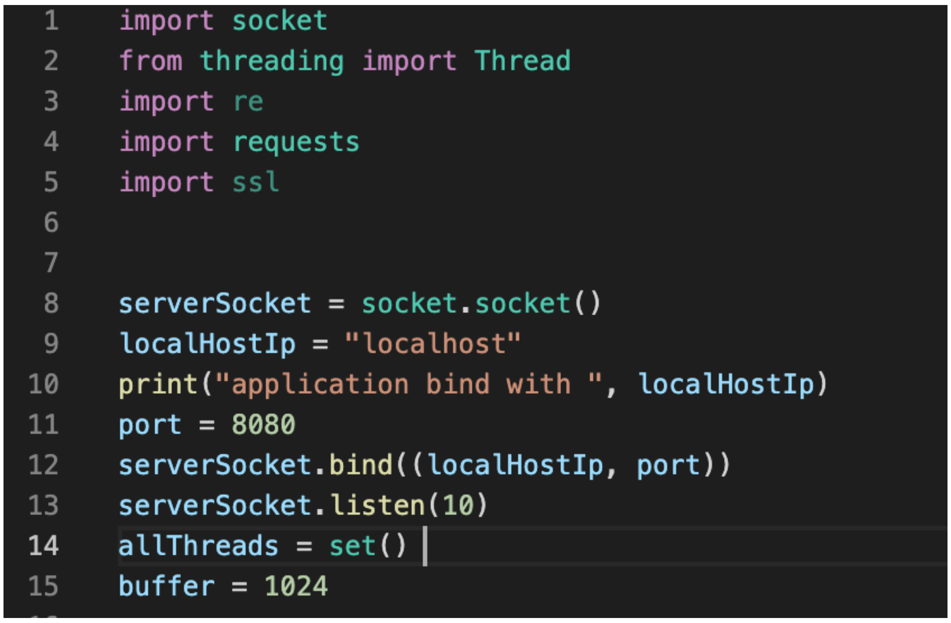
7) close()

When data transmission/reception is no longer required, use it to close the socket.

8) Thread()

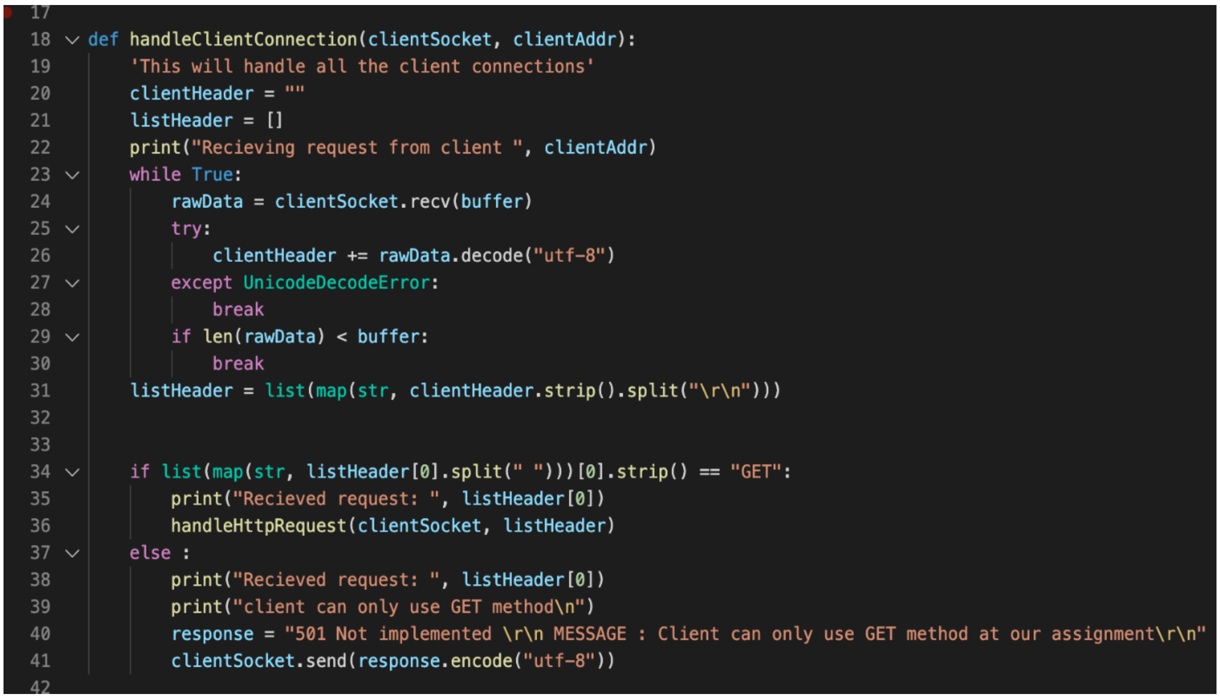
The reason why we use Thread is to send messages from servers and clients in any order. With Thread, you can not only handle one thing at a time in one program, but also handle several things at the same time. The thread() function receives several factors, and the factors used in TCP socket communication are target and args. Target is the part where a function to be executed through threading is input, and the parameters required for the function are transferred to args. However, at this time, if there is one factor in args, '' should be added to recognize it as a tuple. The thread thus generated must start working through the start() function.

4. Implementation Details



```python
import socket
from threading import Thread
import re
import requests
import ssl


serverSocket = socket.socket()
localHostIp = "localhost"
print("application bind with ", localHostIp)
port = 8080
serverSocket.bind((localHostIp, port))
serverSocket.listen(10)
allThreads = set() 
buffer = 1024
```

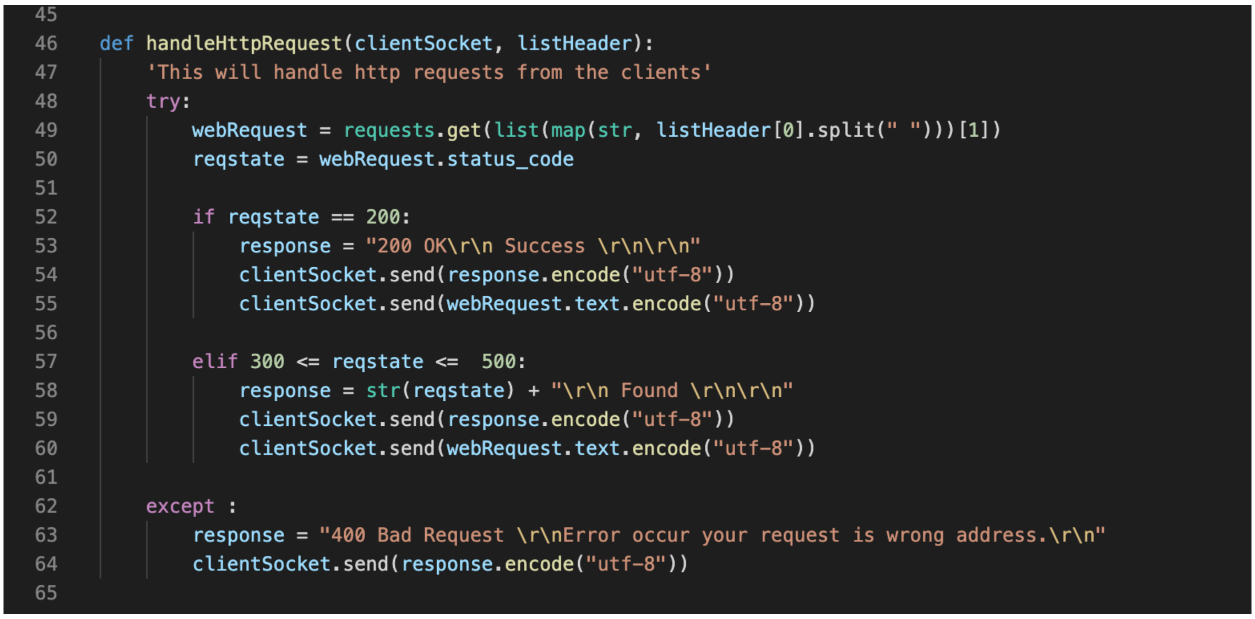
Using socket.socket() I created socket object. And set localHostIp = “localhost”, so work for all ips of the sever computer. Port number set 8080. And bind to server socket ky localhost and port number. Sover.Socket.listen(10) mean can allow connection to 10 clients. allThreads = set() for store records of all threads for joining them. Last, set buffer size of server 1024 bytes.



```python
def handleClientConnection(clientSocket, clientAddr):
    'This will handle all the client connections'
    clientHeader = ""
    listHeader = []
    print("Recieving request from client ", clientAddr)
    while True:
        rawData = clientSocket.recv(buffer)
        try:
            clientHeader += rawData.decode("utf-8")
        except UnicodeDecodeError:
            break
        if len(rawData) < buffer:
            break
    listHeader = list(map(str, clientHeader.strip().split("\r\n")))


    if list(map(str, listHeader[0].split(" ")))[0].strip() == "GET":
        print("Recieved request: ", listHeader[0])
        handleHttpRequest(clientSocket, listHeader)
    else :
        print("Recieved request: ", listHeader[0])
        print("client can only use GET method\n")
        response = "501 Not implemented \r\n MESSAGE : Client can only use GET method at our assignment\r\n"
        clientSocket.send(response.encode("utf-8"))
```

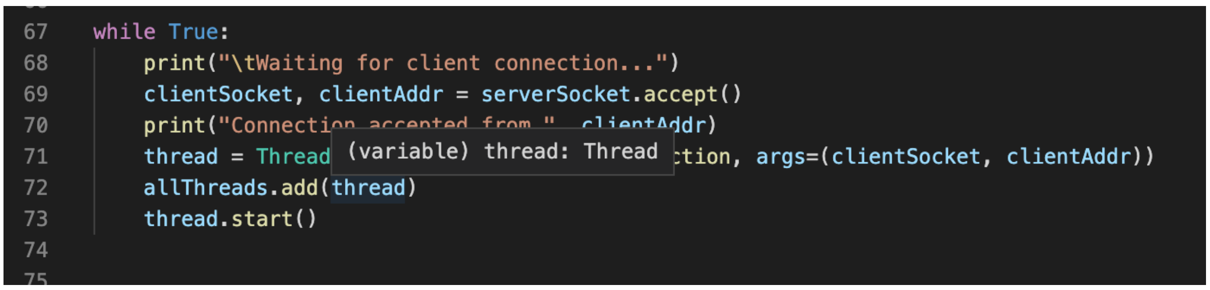
This function handle all the client connection. First, this function print client information about address and port number. And get client's command at while loop. Using socket.recv() receive request from client. And split the command and store at listHeader.

And check the request method is GET or not. If the request method is get then print what is the client command and implement handleHTTPRequest function. If not, print "client can only use GET method". Because at the homework #2, Not Implemented (501) for valid HTTP methods other than GET. So, save the message at respone. And using clientSocket.send() function, sent respone to client.



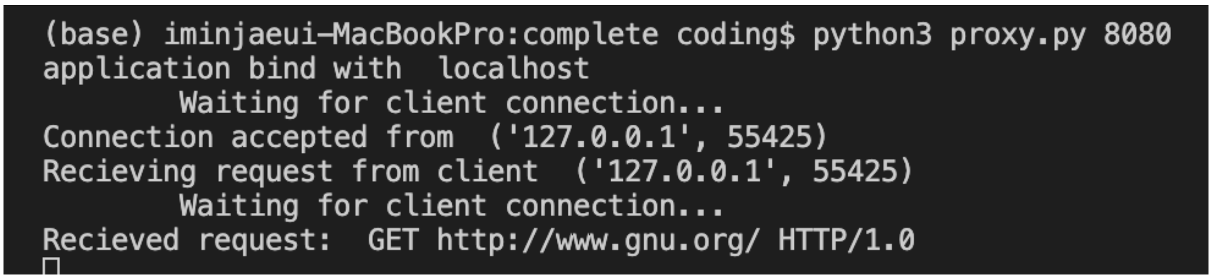
```python

def handleHttpRequest(clientSocket, listHeader):
    'This will handle http requests from the clients'
    try:
        webRequest = requests.get(list(map(str, listHeader[0].split(" ")))[1])
        reqstate = webRequest.status_code

        if reqstate == 200:
            response = "200 OK\r\n Success \r\n\r\n"
            clientSocket.send(response.encode("utf-8"))
            clientSocket.send(webRequest.text.encode("utf-8"))

        elif 300 <= reqstate <=  500:
            response = str(reqstate) + "\r\n Found \r\n\r\n"
            clientSocket.send(response.encode("utf-8"))
            clientSocket.send(webRequest.text.encode("utf-8"))

    except :
        response = "400 Bad Request \r\nError occur your request is wrong address.\r\n"
        clientSocket.send(response.encode("utf-8"))

```

Next function is handleHTTPRequest. This get clinetSocket and listHeader. I use try exception, because when the address is wrong. requests.get function occur error. So. If client's request are wrong address, this function can handle error. Using request.get we can store the respone of origin server at webRequest. webRequest.status_code has the value 1xx~5xx by the state of respone. If the client request success. The status_code value is 200. So I make respone 200 OK and send to clinet by using clientSocket.send, and the origin server's information encoded by text then send to clinet. Send form can various. For example the code written by webRequest.text, content, json, encoding and so on. If the error occur in try: make the respone and sent to client that Error occur your request is wrong address.



```
while True:
    print("\tWaiting for client connection...")
    clientSocket, clientAddr = serverSocket.accept()
    print("Connection accepted from ", clientAddr)
    thread = Thread (variable) thread: Thread ction, args=(clientSocket, clientAddr))
    allThreads.add(thread)
    thread.start()
```

Using serverSocket.accpet() establish connection with client. Creating new thread and then handling client connection to accept another client connection as well.

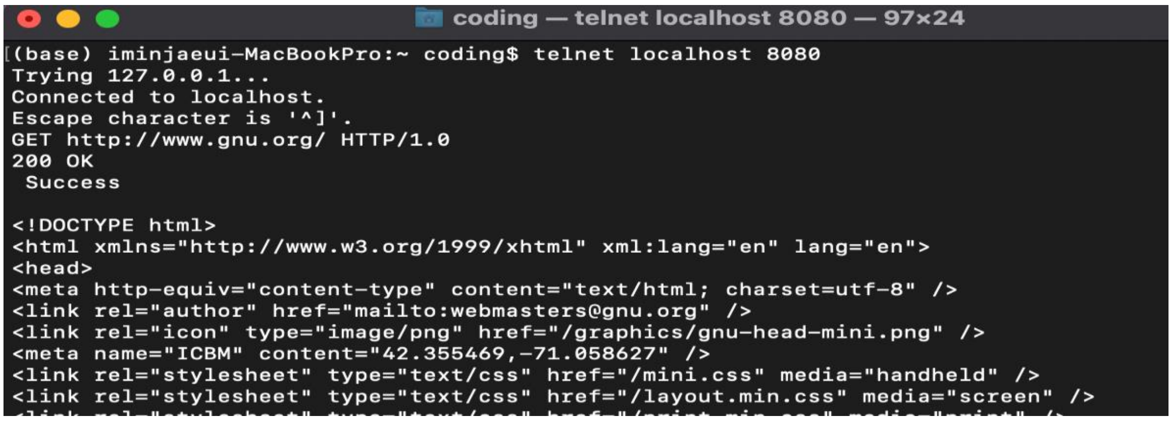


```
(base) iminjaeui-MacBookPro:complete coding$ python3 proxy.py 8080
application bind with  localhost
        Waiting for client connection...
Connection accepted from  ('127.0.0.1', 55425)
Recieving request from client  ('127.0.0.1', 55425)
        Waiting for client connection...
Recieved request:  GET http://www.gnu.org/ HTTP/1.0
```

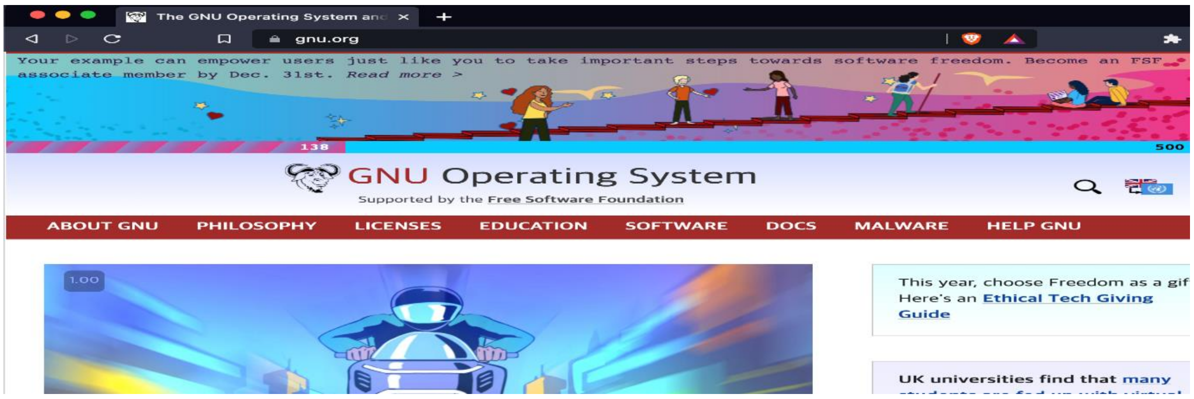
This is terminal picture visual studio code. If I implement the proxy.py, the proxy server waiting for client connection. And command “telnet local host 8080” at cmd. I can access proxy server. Then the terminal show that “Connection accepted from (ip adress, port number)”.If I command “Get http://www.sdkfjadf***” , proxy server recive the request. Then terminal show that “Receiving request from client (ip address, port number)” “Receieved request: “GET http://www.skjdflkja***”.

5. Result

< request example 1 >

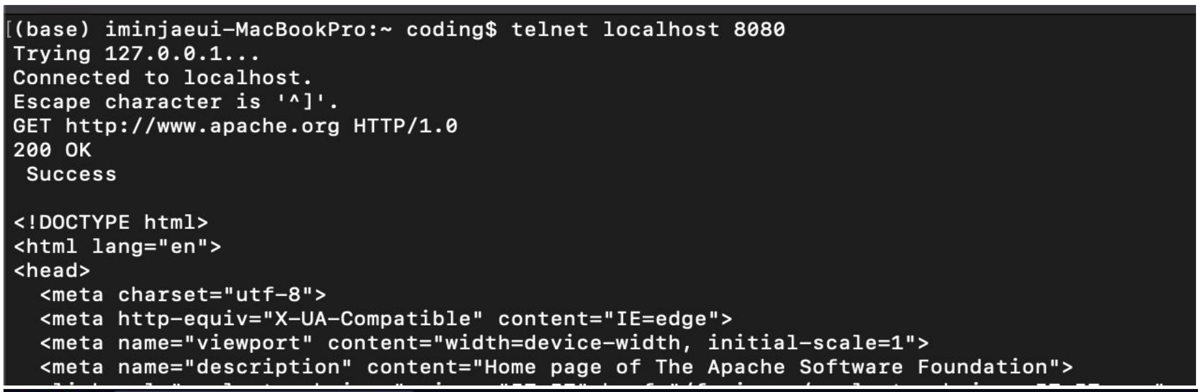


```
(base) iminjaeui-MacBookPro:~ coding$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET http://www.gnu.org/ HTTP/1.0
200 OK
 Success

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link rel="author" href="mailto:webmasters@gnu.org" />
<link rel="icon" type="image/png" href="/graphics/gnu-head-mini.png" />
<meta name="ICBM" content="42.355469,-71.058627" />
<link rel="stylesheet" type="text/css" href="/mini.css" media="handheld" />
<link rel="stylesheet" type="text/css" href="/layout.min.css" media="screen" />
```

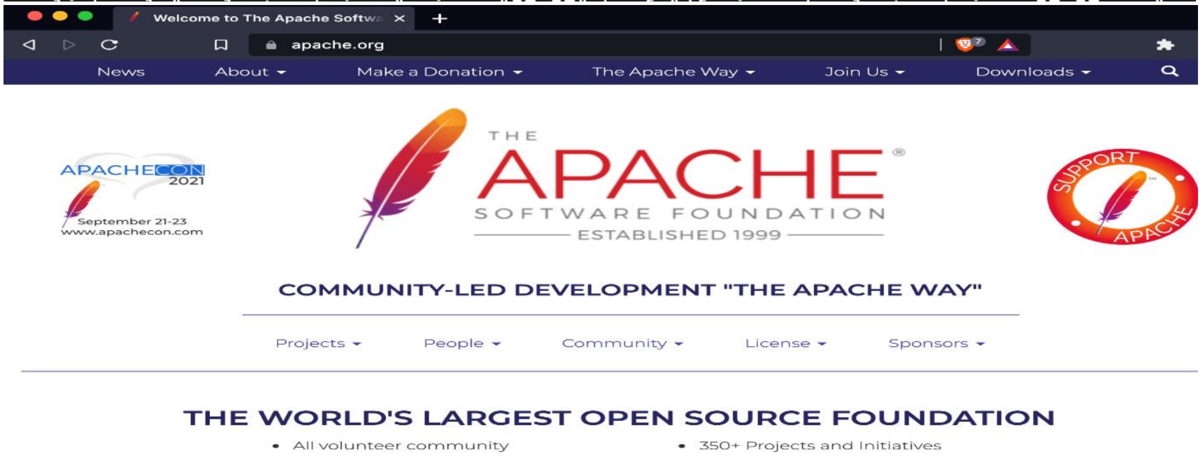

The GNU Operating System and
gnu.org
Your example can empower users just like you to take important steps towards software freedom. Become an FSF associate member by Dec. 31st. Read more >
GNU Operating System
Supported by the Free Software Foundation
ABOUT GNU
PHILOSOPHY
LICENSES
EDUCATION
SOFTWARE
DOCS
MALWARE
HELP GNU
This year, choose Freedom as a gif
Here's an Ethical Tech Giving Guide
UK universities find that many

I request http://www.gnu.org HTTP/1.0 and get the result. 200 OK. I also access using chrome browser and check the cite exist well.

< request example 2 >

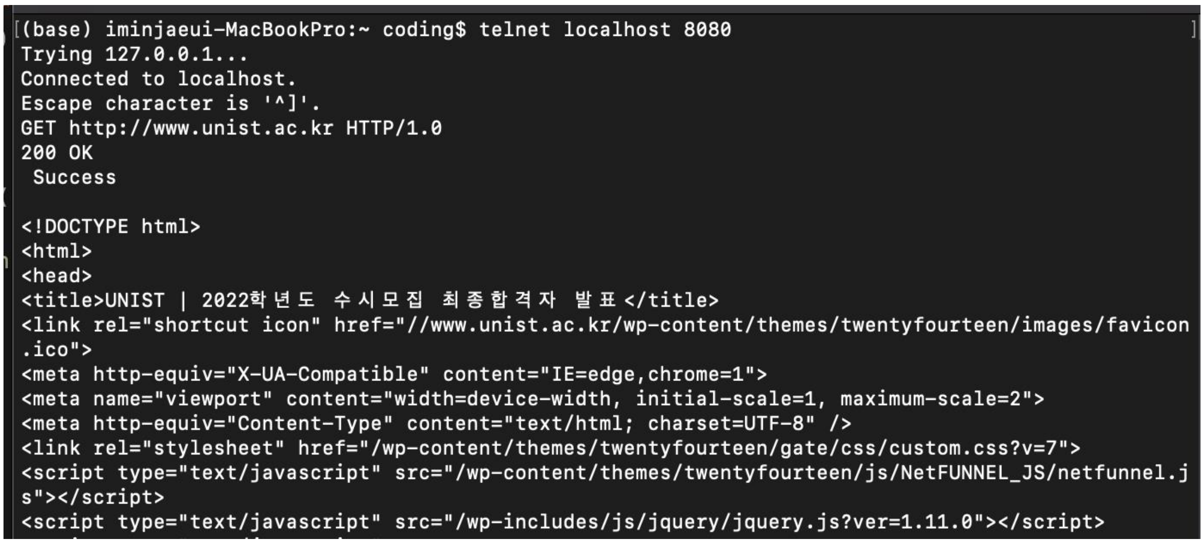


```
[(base) iminjaeui-MacBookPro:~ coding$ telnet localhost 8080
 Trying 127.0.0.1...
 Connected to localhost.
 Escape character is '^]'.
 GET http://www.apache.org HTTP/1.0
 200 OK
  Success

 <!DOCTYPE html>
 <html lang="en">
 <head>
   <meta charset="utf-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <meta name="description" content="Home page of The Apache Software Foundation">
```

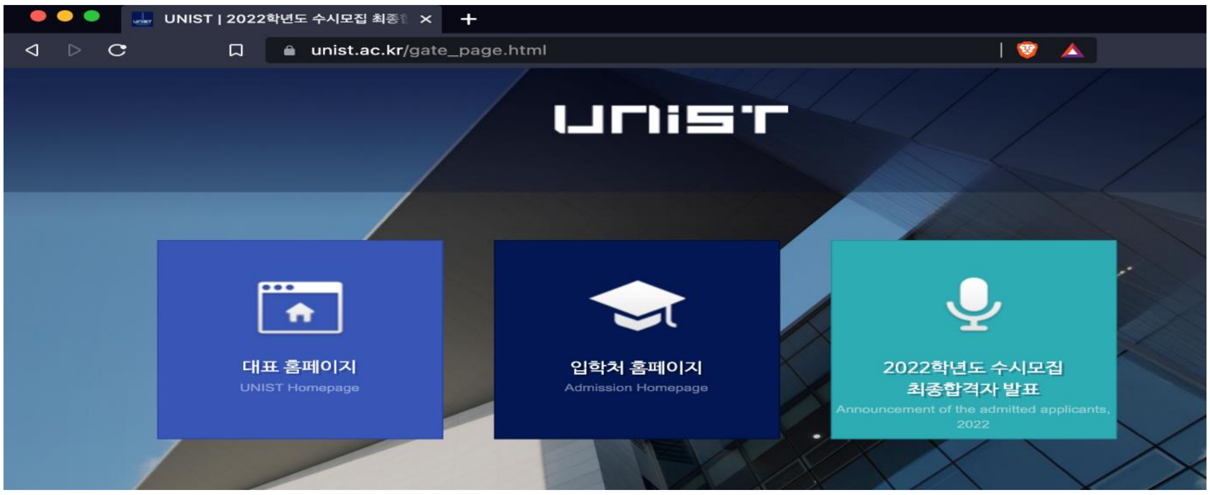


I request http://www.apache.org HTTP/1.0 and get the result. 200 OK. I also access using chrome browser and check the cite exist well. I found this url at https://whynohttps.com/ at the assignment explanation pdf.

< request example 3 >

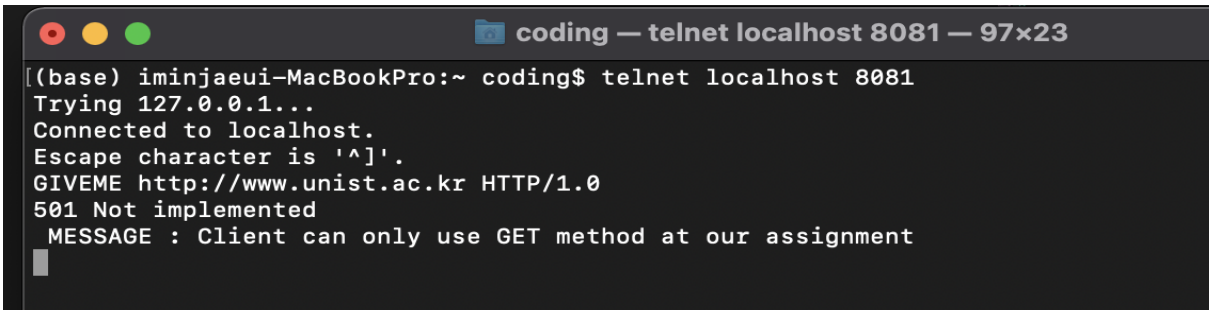


```
[(base) iminjaeui-MacBookPro:~ coding$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET http://www.unist.ac.kr HTTP/1.0
200 OK
 Success

<!DOCTYPE html>
<html>
<head>
<title>UNIST | 2022학년도 수시모집 최종합격자 발표</title>
<link rel="shortcut icon" href="//www.unist.ac.kr/wp-content/themes/twentyfourteen/images/favicon
.ico">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=2">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="stylesheet" href="/wp-content/themes/twentyfourteen/gate/css/custom.css?v=7">
<script type="text/javascript" src="/wp-content/themes/twentyfourteen/js/NetFUNNEL_JS/netfunnel.j
s"></script>
<script type="text/javascript" src="/wp-includes/js/jquery/jquery.js?ver=1.11.0"></script>
```

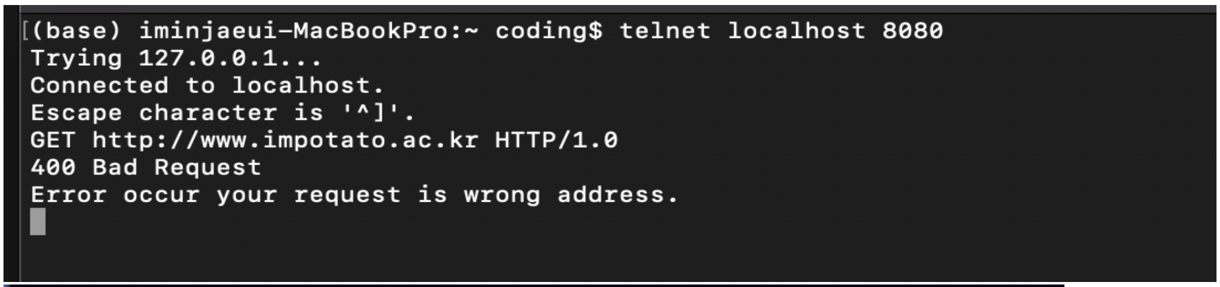

UNIST | 2022학년도 수시모집 최종
unist.ac.kr/gate_page.html
UNIST
대표 홈페이지
UNIST Homepage
입학처 홈페이지
Admission Homepage
2022학년도 수시모집
최종합격자 발표
Announcement of the admitted applicants, 2022

I request http://www.apache.org HTTP/1.0 and get the result. 200 OK. I also access using chrome browser and check the cite exist well. I also check our school's portal.

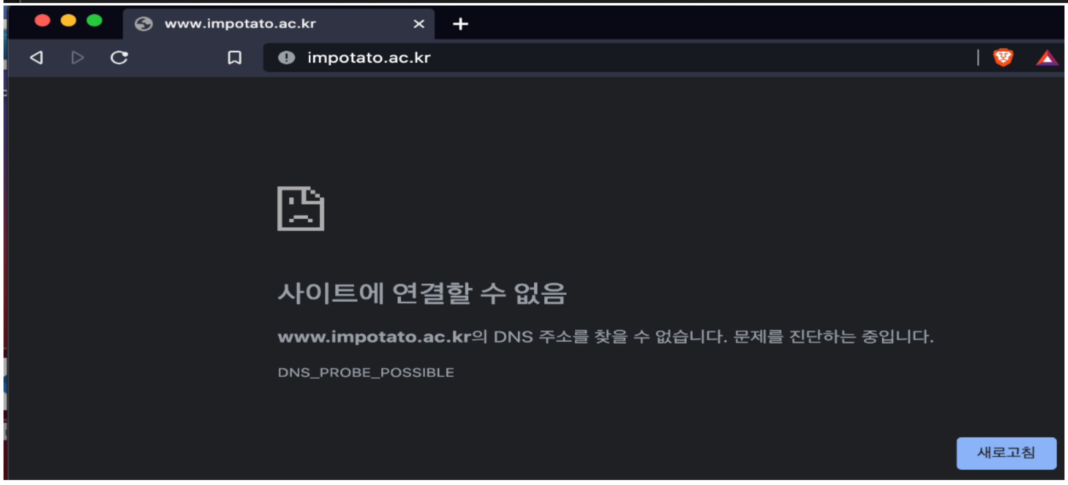
< bad request example >



```
coding — telnet localhost 8081 — 97×23
[(base) iminjaeui-MacBookPro:~ coding$ telnet localhost 8081
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GIVEME http://www.unist.ac.kr HTTP/1.0
501 Not implemented
 MESSAGE : Client can only use GET method at our assignment
```



```
[(base) iminjaeui-MacBookPro:~ coding$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET http://www.impotato.ac.kr HTTP/1.0
400 Bad Request
Error occur your request is wrong address.
```



www.impotato.ac.kr

impotato.ac.kr

사이트에 연결할 수 없음

www.impotato.ac.kr의 DNS 주소를 찾을 수 없습니다. 문제를 진단하는 중입니다.

DNS_PROBE_POSSIBLE

As the assignment request, If I does not use GET method(use GIVEME), the proxy server respone that (501) Not implemented.

And the url that www.impotato.ac.kr is not exist. If I command wrong address, then the proxy server respone that (400) Bad Request.

6. Conclusion

I implemented a simple proxy server with Python. Homework 2 is not require cache and multi I/O, I doesn't make this function. If client request the proxy server, then proxy server request origin server and get respone. And proxy server check the respone that is valid or not and send to clinet. As I tried socket programming myself, I got to know several Python modules, and I looked up examples and learned them through searching. During this assignment, I got to understand in detail the concept of a proxy server that I only knew in theory, and I got used to socket programming that I was not familiar with.