

## Programming Assignment #2: WWW

### Group no 15

Sanket P. Rathod (CS23MTECH11033)

Sanket S. Deone (CS23MTECH11034)

Manas Upadhyaya (SM22MTECH12002)

### ANTI-PLAGIARISM Statement

*We certify that this assignment/report is our own work, based on our personal study and/or research and that we have acknowledged all material and sources used in its preparation, whether they be books, articles, packages, datasets, reports, lecture notes, and any other kind of document, electronic or personal communication. We also certify that this assignment/report has not previously been submitted for assessment/project in any other course lab, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that we have not copied in part or whole or otherwise plagiarized the work of other students and/or persons. Additionally, we acknowledge that we may have used AI tools, such as language models (e.g., ChatGPT, Bard), for assistance in generating and refining my assignment, and we have made all reasonable efforts to ensure that such usage complies with the academic integrity policies set for the course. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, we understand our responsibility to report honour violations by other students if we become aware of it.*

## PART-1 : A Simple Web Client

### Code Explanation:

- Importing packages one for Socket and other for parsing HTML pages.

```
#importing packages
import socket
from bs4 import BeautifulSoup
```

- Taking a Choice from user that he wants to use a proxy server or not

```
#Taking a choice from client that it wants connection through Proxy Server or Direct server
choice = input("Press P if you want connection through proxy Server else press S : ")
```

- If user wants to use a Proxy server to get request from the server.
- Then enter Proxy host at which proxy server is located.
- Enter proxy Port, Server host, port and path.

```
#Enter Proxy host, port, Server host, port and path
proxy_host = input("Enter Proxy Host : ")
proxy_port = int(input("Enter Proxy Port : "))
server_host = input("Enter host name : ")
server_port = int(input("Enter port number :"))
path = input("Enter path : ")
```

- Now create a client socket for sending a get request to the proxy server.
- And connect client socket to the proxy server.
- Create a GET request and send the request to proxy.

```
#Create an Object for Socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
#Making a Connection with the Proxy Server
client_socket.connect((proxy_host, proxy_port))

#Generating A GET request
request = f"GET {path} HTTP/1.1\r\nHost: {server_host}:{server_port}\r\n\r\n"

#Send Request to the Web Proxy
client_socket.send(request.encode())
```

- Receive response coming from the proxy server. And print response then close client socket.

```
#receiving response
response = b''
while True:
    data = client_socket.recv(4096)
    if not data:
        break
    response += data
#Print Received Response
print(response.decode())

#Closing Connection with proxy
client_socket.close()
```

- Now we have to parse Objects.
- We have used BeautifulSoup to parse HTML files from the proxy response.

```
#Using BeautifulSoup for Parsing objects
soup = BeautifulSoup(response.decode(), 'html.parser')
paths = []
for atag in soup.find_all('a'):
    paths.append(atag['href'])
#Filtering out HTML files from the parsed objects
html_paths = [path for path in paths if path.startswith('/') and path.endswith('.html')]
```

- Now for each object we have to create a non persistent connection to fetch objects.

```
#for Fetching Objects One by one using Non-Persistent Connection
i=0
for obj_path in html_paths :
    #initialized object host and port
    obj_host = server_host
    obj_port = server_port

    #Created connection for each object
    obj_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    obj_socket.connect((proxy_host, proxy_port))
```

- Send Get object request to proxy server.

```
#creating request for proxy Server
request = f"GET {obj_path} HTTP/1.1\r\nHost: {obj_host}:{obj_port}\r\n\r\n"

#Sending Request to proxy server
obj_socket.send(request.encode())
```

- Receiving Objects one by one.

```
#Receiving Response from Proxy server
obj_response = b''
while True:
    obj_data = obj_socket.recv(4096)
    if not obj_data:
        break
    obj_response += obj_data
print(obj_response.decode())
```

- Close the connection after receiving 1 object.

- Now if Client wants to connect Server directly.
- Instead of giving proxy host and proxy port just give Server host and server port.

```
#Getting Server host, port and path
server_host = input("Enter host name : ")
server_port = int(input("Enter port number :"))
path = input("Enter path : ")
```

- Now create a client socket for sending a get request to the web server.
- And connect client socket to the web server.
- Create a GET request and send the request to web server.
- Receive response coming from the web server. And print response then close client socket.
- And parse the Objects as fetched in case of Proxy server.

## Code Output: For gaia.cs.umass.edu

- Index Page

```
ubuntu@cs23mtech11033:~/test2$ python3 client.py
Press P if you want connection through proxy Server else press S : S
Enter host name : gaia.cs.umass.edu
Enter port number :80
Enter path : /
HTTP/1.1 200 OK
Date: Sun, 05 Nov 2023 11:36:58 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Tue, 01 Mar 2016 18:57:50 GMT
ETag: "a5b-52d015789ee9e"
Accept-Ranges: bytes
Content-Length: 2651
Content-Type: text/html; charset=UTF-8

<html>
<head>
<title>Computer Network Research Group - UMass Amherst
</title>
</head>
<body bgcolor="#ffffff">
<center>
<p>
<map name="cnrg_imapMAP">
<area coords="290,177,407,205" shape="rect" href="/networks/resources/index.html">
<area coords="163,178,275,205" shape="rect" href="/networks/education/index.html">
<area coords="62,165,145,191" shape="rect" href="/search.html">
<area coords="6,63,157,90" shape="rect" href="/networks/collaborations.html">
<area coords="64,7,146,34" shape="rect" href="/networks/people.html">
<area coords="163,7,270,33" shape="rect" href="/networks/research.html">
<area coords="288,6,417,33" shape="rect"
href="/networks/publications.html">
</map>
<p>
<BR>
<BR>
```

```
ensor networks, wireless networks, and network measurement. We seek a principled understanding of
a complementary mix of theoretical and applied experimental research.
</td>
<td width="10%">&nbsp;</td>
</tr>
</table>

</h4>

</font>
<p>
<!-- QUP Button & QualNet link markup start -->
<a href="http://www.scalable-networks.com/customers/qup/index.php">
</a>
<a href="http://www.scalable-networks.com">QualNet Network Simulator</a>
<!-- End of QUP Button HTML & QualNet link markup -->
<p>
<p>
<center>
<font size="1">
<a href="/networks/people.html"> PEOPLE |</a>
<a href="/networks/research.html"> RESEARCH |</a>
<a href="http://www-net.cs.umass.edu/networks/publications.html"> PUBLICATIONS |</a>
<a href="/networks/collaborations.html"> COLLABORATIONS |</a>
<a href="http://gaia.cs.umass.edu/search.html"> SEARCH |</a>
<a href="/networks/education/index.html"> EDUCATION |</a>
<a href="/networks/resources/index.html">
RESOURCES</a>
<p>

</font>
</center>
</body>
```

There is a more content in the index file that we have received but we have not showed it here because it's very large output.

- Object 1

From here we get objects using Non-Persistent connection.

```
HTTP/1.1 200 OK
Date: Sun, 05 Nov 2023 11:37:04 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Tue, 01 Mar 2016 18:57:50 GMT
ETag: "a5b-52d015789ee9e"
Accept-Ranges: bytes
Content-Length: 2651
Content-Type: text/html; charset=UTF-8

<html>
<head>
<title>Computer Network Research Group - UMass Amherst
</title>
</head>
<body bgcolor="#ffffff">
<center>
<p>
<map name="cnrg_imapMAP">
<area coords="290,177,407,205" shape="rect" href="/networks/resources/index.html">
<area coords="163,178,275,205" shape="rect" href="/networks/education/index.html">
<area coords="62,165,145,191" shape="rect" href="/search.html">
<area coords="6,63,157,90" shape="rect" href="/networks/collaborations.html">
<area coords="64,7,146,34" shape="rect" href="/networks/people.html">
<area coords="163,7,270,33" shape="rect" href="/networks/research.html">
<area coords="288,6,417,33" shape="rect"
href="/networks/publications.html">
</map>
<p>
<BR>

</h4>

</font>
<p>
<!-- QUP Button & QualNet link markup start -->
<a href="http://www.scalable-networks.com/customers/qup/index.php">
</a>
<a href="http://www.scalable-networks.com">QualNet Network Simulator</a>
<!-- End of QUP Button HTML & QualNet link markup -->
<p>
<p>
<center>
<font size="1">
<a href="/networks/people.html"> PEOPLE |</a>
<a href="/networks/research.html"> RESEARCH |</a>
<a href="http://www-net.cs.umass.edu/networks/publications.html"> PUBLICATIONS |</a>
<a href="/networks/collaborations.html"> COLLABORATIONS |</a>
<a href="http://gaia.cs.umass.edu/search.html"> SEARCH |</a>
<a href="/networks/education/index.html"> EDUCATION |</a>
<a href="/networks/resources/index.html">
RESOURCES</a>
<p>

</font>
</center>
</body>
</html>

.....Received Obj 1 .....
```

- Object 2



```

HTTP/1.1 200 OK
Date: Sun, 05 Nov 2023 11:37:15 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Thu, 05 Apr 2018 11:24:44 GMT
ETag: "46d7-569182fb67ea5"
Accept-Ranges: bytes
Content-Length: 18135
Content-Type: text/html; charset=UTF-8

<html><head><title>People</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"></head>

<body><table border="0" cellpadding="0" cellspacing="0" width="856">
  <tbody>
    <tr>
      <td height="19" width="551"></td>
      <td width="299"></td>
      <td width="1"></td>
      <td width="5"></td>
    </tr>
    <tr>
      <td height="224" valign="top">
        <p align="center"><br>
        </p>
        <p align="left"> <font face="Arial, Helvetica, sans-serif" size="7">People</font></p>
        <p align="left"><br>
        <font face="Arial, Helvetica, sans-serif"><i><font size="2">"The people make the town ."  

        Greek proverb</i></font></p>
        </td>
      <td height="224" valign="top"><br>
      
      <map name="cnrg_imapMAP">
      <area coords="290,177,407,205" shape="rect" href="http://gaia.cs.umass.edu/networks/resources/index.html">

      <li><font face="Arial, Helvetica, sans-serif"><a href="http://people.cs.umass.edu/~jenstesh/">Jennie Steshenko</a></font></li>
      <li><font face="Arial, Helvetica, sans-serif"><b><a href="http://www-net.cs.umass.edu/%7Ekwsuh">Kyoungwon Suh, </a></b>
      now at Illinois State University </font></li>
      <li><font face="Arial, Helvetica, sans-serif"><b><a href="http://www.cs.umass.edu/%7Ebmg">Bing Wang</a></b> now at the
      <a href="http://www.cse.uconn.edu/">Computer Science and Engineering Department</a>,
      University of Connecticut.</font></li>
      <li><font face="Arial, Helvetica, sans-serif"><b> Maya
      Yajnik</b>, now at <a href="http://www.telcordia.com/">Telcordia </a> </font></li>
      <li><font face="Arial, Helvetica, sans-serif"><a href="http://www.cs.umass.edu/%7Eshyang">Sookhyun Yang</a></font></li>
      <li><font face="Arial, Helvetica, sans-serif"><b><a href="http://www.bentley.edu/academics_research/faculty_research/faculty_database/faculty_detail.cfm?id=1150639">David Yates</a></b></font>, <font face="Arial, Helvetica, sans-serif">now at Bentley College </font></li>
      <li><font face="Arial, Helvetica, sans-serif"><b><a href="http://gaia.cs.umass.edu/networks/domino.research.ibm.com/comm/research_people.nsf/pages/czhang1.index.html">Chun Zhang</a></b></font>, <font face="Arial, Helvetica, sans-serif">now at IBM Research </font></li>
      <li><font face="Arial, Helvetica, sans-serif"><b><a href="http://www.suffolk.edu/college/16414.html">Honggang Zhang</a></b></font>, <font face="Arial, Helvetica, sans-serif">now at Suffolk University </font></li>
      <li><font face="Arial, Helvetica, sans-serif"><b><a href="http://www.cs.umass.edu/%7Eellenz">Ellen(XiaoLan) Zhang</a></b></font>, <font face="Arial, Helvetica, sans-serif">now at Fordham University </font></li>
      <li><font face="Arial, Helvetica, sans-serif"><b><a href="http://www.cs.umn.edu/%7Ezhzhang/home.html">Zhi-Li Zhang</a></b>,
      now at the Computer Science Department, University of Minnesota.</font></li>
    </ul>
    <p>&nbsp;</p>
    <p><i><font face="Arial, Helvetica, sans-serif" size="2">August 2014</font></i></p>
  </td>
</tr>
</tbody>
</table>

</body></html>

.....Received Obj 3 .....

```

- Object 4



```

HTTP/1.1 200 OK
Date: Sun, 05 Nov 2023 11:37:20 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Tue, 27 Jan 2004 21:54:24 GMT
ETag: "ab8-3d1ecede7a400"
Accept-Ranges: bytes
Content-Length: 2744
Content-Type: text/html; charset=UTF-8

<HTML>
<HEAD>
<TITLE>UMass Computer Networks Research Group Research</TITLE>

<BODY BACKGROUND="images/bg.jpg" MARGINHEIGHT=3 MARGINWIDTH=2 TOPMARGIN=0
LEFTMARGIN=0 RIGHTMARGIN=5
BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#5B69A6" VLINK="#5B69A6"
ALINK="#00FF00"
>
<TABLE BORDER=0 CELLPADDING=0 CELLSPACING=0 WIDTH="700" ALIGN=left>
  <TR VALIGN=top>
    <TD ALIGN=right WIDTH=190>  </TD>
    <TD ALIGN=center WIDTH=500>
      <A NAME="top"> <IMG SRC="images/banner-research.jpg" BORDER=0 HSPACE=5> </A>
    </TD>
  </TR>
  <TR VALIGN=top>
    <TD ALIGN=left WIDTH=18%>
      <TABLE BORDER=0 CELLPADDING=0 CELLSPACING=0 ALIGN=left>
        <tr> <td>
          <A HREF="index.html" >
            <IMG SRC="images/homebutton.jpg" BORDER=0></A>

```

```

<ul>
<li> <a href="http://www.casa.umass.edu"> CASA: </a>Collaborative Adaptive Sensing of the Atmosphere <br>
<li> <a href="http://gaia.cs.umass.edu/minc/">
  MINC: </a> Multicast-based Inference of Network-Internal Characteristics<br>
<li><a href="cnrg_AMPS1.pdf">AMPS:</a> Active Multimedia Proxy Services <br>
<li><a href="http://gaia.cs.umass.edu/~yongliu/fluid.html">
  Fluid models for large heterogeneous networks</a><br>
<li><a href="cnrg_matchmaker_10_02_v2.pdf"> Publish-Subscribe networks </a>
<li><a href="cnrg_wireless-10-02.pdf"> Wireless networks </a>
<li><a href="cnrg_measurement-10-02.pdf"> Network Measurement </a>
<li><a href="http://gaia.cs.umass.edu/fluidsimsim/">
  Fluid Simulation </a><br>
<li><a href="http://manic.cs.umass.edu/">
  MANIC:</a> Multimedia Asynchronous Networked Individualized Courseware

</ul>

Copies of our publications can be found at the CNRG
<a href="http://gaia.cs.umass.edu/networks/publications.html">
publications page. </a>

</tr>
</table>

</BODY>
</HEAD>
</HTML>

```

```

.....Received Obj 4 .....

```

Code Output: For [www.example.com](http://www.example.com)

```

ubuntu@cs23mtech11033:~/test2$ python3 client.py
Press P if you want connection through proxy Server else press S : S
Enter host name : www.example.com
Enter port number :80
Enter path : /
HTTP/1.1 200 OK
Accept-Ranges: bytes
Age: 180147
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Sun, 05 Nov 2023 11:28:49 GMT
Etag: "3147526947"
Expires: Sun, 12 Nov 2023 11:28:49 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (nyb/1D1C)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256

<!doctype html>
<html>
<head>
  <title>Example Domain</title>

  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">
    body {
      background-color: #f0f0f2;
      margin: 0;
      padding: 0;
      font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial,
sans-serif;
    }
    div {

```

```

      width: 600px;
      margin: 5em auto;
      padding: 2em;
      background-color: #fdfdff;
      border-radius: 0.5em;
      box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
      color: #38488f;
      text-decoration: none;
    }
    @media (max-width: 700px) {
      div {
        margin: 0 auto;
        width: auto;
      }
    }
  </style>
</head>

<body>
<div>
  <h1>Example Domain</h1>
  <p>This domain is for use in illustrative examples in documents. You may use this
domain in literature without prior coordination or asking for permission.</p>
  <p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>

```

## Code Output: For [www.iith.ac.in](http://www.iith.ac.in)

- In case of [www.iith.ac.in](http://www.iith.ac.in)

- It uses https, and it is a secured version of http so we get output here as 301 Permanently Moved.

```
ubuntu@cs23mtech11033:~/test2$ python3 client.py
Press P if you want connection through proxy Server else press S : S
Enter host name : www.iith.ac.in
Enter port number :80
Enter path : /
HTTP/1.1 301 Moved Permanently
Date: Sun, 05 Nov 2023 12:33:59 GMT
Server: Apache
Location: https://www.iith.ac.in/
Content-Length: 231
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="https://www.iith.ac.in/">here</a>.</p>
</body></html>
```

## **PART-2: A Simple Web Proxy Server**

## Code Explanation:

- Importing Packages

```
#importing packages
import socket
import ssl
import threading
import time
```

- Assigning host id and port to the web proxy and Creating Sockets to communicating with the server.

```
#Assigning Host and port to Proxy
proxy_host = '127.0.0.1'
proxy_port = 9696

#creating Socket for communicating with server
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
#binding Socket with proxy host and port
server.bind((proxy_host, proxy_port))
server.listen(5)

print(f"Proxy server listening on {proxy_host}:{proxy_port}")
```

- Start accepting requests

```
while True:
    #Accepting requests
    client_sock, address = server.accept()
```

- Storing start time so that we can identify how much time required to get request.

```
#Store Starting time
i = i+1
if (flag == 0):
    start_time = time.time()
    flag = 1
```

- Creating thread to handle multiple clients. After that Store end time and calculate total time required to get request.

```
print(f"Accepted connection from {address[0]}:{address[1]}")

#Create thread to handle multiple clients
multiple_clients = threading.Thread(target=handle_client, args=(client_sock,))
multiple_clients.start()

#Storing Ending time
end_time = time.time()

#Calculating total time required to get Response
total_time = (end_time - start_time)*1000
```

- Receive request from the client. And Extract that request that is get server host and server port.

```

try:

    #receiving Client request
    request = client_sock.recv(4096)

    #calling function to extract request coming from the client
    if request:
        target_host,target_port = extract_host(request)
        print(f"Target Host = {target_host}")

```

```

def extract_host(request):
    # Extracting the Header

    #splitting the Request to get host and port from request
    for line in request.split(b'\n'):
        if line.startswith(b'Host:'):
            host_line = line.split(b' ')[1].strip()
            host_parts = host_line.split(b':')

            if len(host_parts) == 2:
                host, port = host_parts
                return host, int(port)
            else:
                #if Request doesnt contain port then default port set to 80
                host = host_parts[0]
                return host, 80

```

- Now create a socket to Communicating with server

```

if target_host:

    #Create Socket
    server_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Connecting to the server host and server port
    server_sock.connect((target_host, target_port))

```

- Sending and receiving response from the server. And send that response back to client.

```

#sending request to the web server
server_sock.send(request)

#Receiving the web server response
while True:

    server_response = server_sock.recv(4096)

    if len(server_response) == 0:
        break
    #Sending Response to the client
    client_sock.send(server_response)

```

- Handling Exceptions

```

#Handling Exception
except ConnectionResetError as connection:
    print(f"Client connection reset: {connection}\n")

except socket.error as soc_error:
    print(f"Socket error: {soc_error}\n")

except Exception as exc_error:
    print(f"An error occurred: {exc_error}\n")




finally:
    #Closing Connection
    client_sock.close()
    server_sock.close()

```

- **References Used :**

[https://gaia.cs.umass.edu/kurose\\_ross/programming/Python\\_code\\_only/Web\\_Proxy\\_programming\\_only.pdf](https://gaia.cs.umass.edu/kurose_ross/programming/Python_code_only/Web_Proxy_programming_only.pdf)

- **To split the request**

 Sure, here's the code with the modifications to specify the desired proxy host and port:  

```

python
Copy code

import socket
import threading

def handle_client(client_socket, target_host, target_port):
    request = client_socket.recv(4096) # Receive the client's request
    if request:
        # Parse the request to extract the target URL
        request_lines = request.split(b'\n')
        first_line = request_lines[0]
        target_url = first_line.split(b' ')[1]

        # Create a socket to connect to the target server
        server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        server_socket.connect((target_host, target_port))

        # Forward the request to the target server
        server_socket.send(request)

```

**Code Output: For gaia.cs.umass.edu**

- **Proxy Side :**
- Here we get Server host and time required to fetch an object.

```
ubuntu@cs23mtech11033:~/test2$ python3 proxy.py
Proxy server listening on 127.0.0.1:9696
Accepted connection from 127.0.0.1:42476
Target Host = b'gaia.cs.umass.edu'
Total time till object1: 1.0704994201660156 msec

Accepted connection from 127.0.0.1:54284
Target Host = b'gaia.cs.umass.edu'
Total time till object2: 5503.95655632019 msec

Accepted connection from 127.0.0.1:54286
Target Host = b'gaia.cs.umass.edu'
Total time till object3: 10993.203401565552 msec

Accepted connection from 127.0.0.1:44448
Total time till object4: 16482.314825057983 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:44460
Total time till object5: 21972.46789932251 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:36438
Total time till object6: 27461.520195007324 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:36454
Total time till object7: 32949.25880432129 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:45066
Total time till object8: 38438.49277496338 msec
```

- **Client Side:**

- Here we need to give Proxy host and proxy port so that client first send request to the proxy and then proxy will send that request to the server.

Index file -

```
ubuntu@cs23mtech11033:~/test2$ python3 client.py
Press P if you want connection through proxy Server else press S : P
Enter Proxy Host : 127.0.0.1
Enter Proxy Port : 9696
Enter host name : gaia.cs.umass.edu
Enter port number : 80
Enter path : /
HTTP/1.1 200 OK
Date: Sun, 05 Nov 2023 14:49:28 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Tue, 01 Mar 2016 18:57:50 GMT
ETag: "a5b-52d015789ee9e"
Accept-Ranges: bytes
Content-Length: 2651
Content-Type: text/html; charset=UTF-8

<html>
<head>
<title>Computer Network Research Group - UMass Amherst
</title>
</head>
<body bgcolor="#ffffff">
<center>
<p>
<map name="cnrg_imapMAP">
<area coords="290,177,407,205" shape="rect" href="/networks/resources/index.html">
<area coords="163,178,275,205" shape="rect" href="/networks/education/index.html">
<area coords="62,165,145,191" shape="rect" href="/search.html">
<area coords="6,63,157,90" shape="rect" href="/networks/collaborations.html">
<area coords="64,7,146,34" shape="rect" href="/networks/people.html">
<area coords="163,7,270,33" shape="rect" href="/networks/research.html">
<area coords="288,6,417,33" shape="rect"
href="/networks/publications.html">

</h4>

</font>
<p>
<!-- QUP Button & QualNet link markup start -->
<a href="http://www.scalable-networks.com/customers/qup/index.php">
</a>
<a href="http://www.scalable-networks.com">QualNet Network Simulator</a>
<!-- End of QUP Button HTML & QualNet link markup -->
<p>
<p>
<center>
<font size="1">
<a href="/networks/people.html"> PEOPLE |</a>
<a href="/networks/research.html"> RESEARCH |</a>
<a href="http://www-net.cs.umass.edu/networks/publications.html"> PUBLICATIONS |</a>
<a href="/networks/collaborations.html"> COLLABORATIONS |</a>
<a href="http://gaia.cs.umass.edu/search.html"> SEARCH |</a>
<a href="/networks/education/index.html"> EDUCATION |</a>
<a href="/networks/resources/index.html">
RESOURCES</a>
<p>

</font>
</center>
</body>
</html>
```



- Object 2

```
HTTP/1.1 200 OK
Date: Sun, 05 Nov 2023 11:37:09 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Tue, 16 Feb 2021 14:43:18 GMT
ETag: "12f4-5bb751f93cdc2"
Accept-Ranges: bytes
Content-Length: 4852
Content-Type: text/html; charset=UTF-8

<!DOCTYPE public "-//w3c//dtd html 4.0 transitional//en"><html><head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <meta name="GENERATOR" content="Mozilla/4.79 [en] (Windows NT 5.0; U) [Netscape]"><title>Don Towsley's homepage</title></head>

  <body bgcolor="#ffffff">
    <p>
    <br> <table border="0" width="75%">
    <tbody><tr>
    <td valign="top" width="149" height="152"></td>

    <td valign="top" rowspan="2" width="395"><b><font size="+2">Don Towsley</font></b>

    <br><font size="+0">College of Information & Computer Sciences </font>
    <br><font size="+0">University of Massachusetts </font>
    <br><font size="+0">Amherst MA 01003 USA </font>
    <br><i><font size="+0"><b>towsley</b> at cs dot umass dot edu</font></i>
    <br><font size="+0">phone: 413-545-0207</font>
    <br><font size="+0">FAX: 413-545-1249</font></td>
    </tr>

    <tr>
    </tr>

  </body>
</html>
```

```
</p><p>Prof. Towsley has received the 2007 <a href="http://www.ieee.org/portal/pages/about/awards/sums/koji.html">
IEEE Koji Kobayashi Computer and Communications Award </a>,
the <a href="http://www.sigmetrics.org/awards.shtml"> 2007 ACM SIGMETRICS Achievement
Award</a>, the <a href="http://www.sigcomm.org/about/awards/sigcomm-awards">2008
ACM SIGCOMM Award</a>, and the 2011 INFOCOM Achievement Award. In addition, he has received numerous paper awards including the
1999 <a href="http://www.comsoc.org/~awards/bennett.html">
IEEE Communications Society William Bennett Award</a>,
the 2008 <a href="http://www.sigcomm.org/awards/test-of-time-paper-award"> ACM SIGCOMM Test of Time Paper Award</a>, the 2012 <a
href="https://www.sigmetrics.org/awards.shtml#testoftime"> ACM SIGMETRICS Test of Time Award</a>, a 2018 ACM MOBICOM Test of Ti
me Award,
and several conference and workshop best paper awards including five ACM SIGMETRICS Awards.
He is also the recipient of the UMass Award for Outstanding Accomplishments in Research and Creative Activity, the University of
Massachusetts Chancellor's Medal, the UMass Amherst Distinguished Graduate Mentor Award, and
an Outstanding Research Award from the College
of Natural Science and Mathematics at the University of Massachusetts.
He has twice received IBM Faculty Fellowship Awards, and is a Fellow of the IEEE and the ACM. Last, he was elected a correspon
ding member of the Brazilian Academy of Sciences.
</p>

<!--<p><font size="+1"><a href="http://gaia.cs.umass.edu/">[Research Group] [</a><a href="http://gaia.cs.umass.edu/papers/papers
.html">Publications]</a><a href="http://gaia.cs.umass.edu/personnel/jkcourses.html"> [Courses]</a></font> <a href="http://gaia.
cs.umass.edu/kurose/talks"><font size="4">[Talks]</font></a><br>
<font size="+1"> -->
</body></html>
```

```
.....Received Obj 2 .....
```

- Object 3

```

HTTP/1.1 200 OK
Date: Sun, 05 Nov 2023 11:37:15 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Thu, 05 Apr 2018 11:24:44 GMT
ETag: "46d7-569182fb67ea5"
Accept-Ranges: bytes
Content-Length: 18135
Content-Type: text/html; charset=UTF-8

<html><head><title>People</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"></head>

<body><table border="0" cellpadding="0" cellspacing="0" width="856">
  <tbody>
    <tr>
      <td height="19" width="551"></td>
      <td width="299"></td>
      <td width="1"></td>
      <td width="5"></td>
    </tr>
    <tr>
      <td height="224" valign="top">
        <p align="center"><br>
        </p>
        <p align="left"> <font face="Arial, Helvetica, sans-serif" size="7">People</font></p>
        <p align="left"><br>
        <font face="Arial, Helvetica, sans-serif"><i><font size="2">"The people make the town ."</font>
        Greek proverb</i></font></p>
        </td>
      <td height="224" valign="top"><br>
      
      <map name="cnrg_imapMAP">
      <area coords="290,177,407,205" shape="rect" href="http://gaia.cs.umass.edu/networks/resources/index.html">

<li><font face="Arial, Helvetica, sans-serif"><a href="http://people.cs.umass.edu/~jenstesh/">Jennie Steshenko</a></font></li>
<li><font face="Arial, Helvetica, sans-serif"><b><a href="http://www-net.cs.umass.edu/%7Ekwsuh">Kyoungwon Suh, </a></b>
now at Illinois State University </font></li>
<li><font face="Arial, Helvetica, sans-serif"><b><a href="http://www.cs.umass.edu/%7Ebing">Bing Wang</a></b> now at the
<a href="http://www.cse.uconn.edu/">Computer Science and Engineering Department</a>,
University of Connecticut.</font></li>
<li><font face="Arial, Helvetica, sans-serif"><b> Maya
Yajnik</b>, now at <a href="http://www.telcordia.com/"> Telcordia </a> </font></li>
<li><font face="Arial, Helvetica, sans-serif"><a href="http://www.cs.umass.edu/%7Eshyang">Sookhyun Yang</a></font></li>
>
<li><font face="Arial, Helvetica, sans-serif"><b><a href="http://www.bentley.edu/academics_research/faculty_research/faculty_database/faculty_detail.cfm?id=1150639">David Yates</a></b></font>, <font face="Arial, Helvetica, sans-serif">now at Bentley College </font></li>
<li><font face="Arial, Helvetica, sans-serif"><b><a href="http://gaia.cs.umass.edu/networks/domino.research.ibm.com/comm/research_people.nsf/pages/czhang1.index.html">Chun Zhang</a></b></font>, <font face="Arial, Helvetica, sans-serif">now at IBM Research </font></li>
<li><font face="Arial, Helvetica, sans-serif"><b><a href="http://www.suffolk.edu/college/16414.html">Honggang Zhang</a></b></font>, <font face="Arial, Helvetica, sans-serif">now at Suffolk University </font></li>
<li><font face="Arial, Helvetica, sans-serif"><b><a href="http://www.cs.umass.edu/%7Eellenz">Ellen(XiaoLan) Zhang</a></b></font>, <font face="Arial, Helvetica, sans-serif">now at Fordham University </font></li>
<li><font face="Arial, Helvetica, sans-serif"><b><a href="http://www.cs.umn.edu/%7Ezhzhang/home.html">Zhi-Li Zhang</a></b></font>,
now at the Computer Science Department, University of Minnesota.</font></li>
</ul>
<p>&nbsp;</p>
<p><i><font face="Arial, Helvetica, sans-serif" size="2">August 2014</font></i></p>
</td>
</tr>
</tbody>
</table>

</body></html>

.....Received Obj 3 .....

```

- Object 4

```

HTTP/1.1 200 OK
Date: Sun, 05 Nov 2023 11:37:20 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Tue, 27 Jan 2004 21:54:24 GMT
ETag: "ab8-3d1ecede7a400"
Accept-Ranges: bytes
Content-Length: 2744
Content-Type: text/html; charset=UTF-8

<HTML>
<HEAD>
<TITLE>UMass Computer Networks Research Group Research</TITLE>

<BODY BACKGROUND="images/bg.jpg" MARGINHEIGHT=3 MARGINWIDTH=2 TOPMARGIN=0
LEFTMARGIN=0 RIGHTMARGIN=5
BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#5B69A6" VLINK="#5B69A6"
ALINK="#00FF00"
>
<TABLE BORDER=0 CELLPADDING=0 CELLSPACING=0 WIDTH="700" ALIGN=left>
  <TR VALIGN=top>
    <TD ALIGN=right WIDTH=190>  </TD>
    <TD ALIGN=center WIDTH=500>
      <A NAME="top"> <IMG SRC="images/banner-research.jpg" BORDER=0 HSPACE=5> </A>
    </TD>
  </TR>
  <TR VALIGN=top>
    <TD ALIGN=left WIDTH=18%>
      <TABLE BORDER=0 CELLPADDING=0 CELLSPACING=0 ALIGN=left>
        <tr> <td>
          <A HREF="index.html" >
            <IMG SRC="images/homebutton.jpg" BORDER=0></A>

```

```

<ul>
<li> <a href="http://www.casa.umass.edu"> CASA: </a>Collaborative Adaptive Sensing of the Atmosphere <br>
<li> <a href="http://gaia.cs.umass.edu/minc/">
  MINC: </a> Multicast-based Inference of Network-Internal Characteristics<br>
<li><a href="cnrg_AMPS1.pdf">AMPS:</a> Active Multimedia Proxy Services <br>
<li><a href="http://gaia.cs.umass.edu/~yongliu/fluid.html">
  Fluid models for large heterogeneous networks</a><br>
<li><a href="cnrg_matchmaker_10_02_v2.pdf"> Publish-Subscribe networks </a>
<li><a href="cnrg_wireless-10-02.pdf"> Wireless networks </a>
<li><a href="cnrg_measurement-10-02.pdf"> Network Measurement </a>
<li><a href="http://gaia.cs.umass.edu/fluidsimsim/">
  Fluid Simulation </a><br>
<li><a href="http://manic.cs.umass.edu/">
  MANIC:</a> Multimedia Asynchronous Networked Individualized Courseware

</ul>

Copies of our publications can be found at the CNRG
<a href="http://gaia.cs.umass.edu/networks/publications.html">
publications page. </a>

</tr>
</table>

</BODY>
</HEAD>
</HTML>

```

.....Received Obj 4 .....

Similarly, there are 7 objects fetched by the client via proxy Server.

**Code Output: For [www.example.com](http://www.example.com)**

- Proxy Side :

```

ubuntu@cs23mtech11033:~/test2$ python3 proxy.py
Proxy server listening on 127.0.0.1:9697
Accepted connection from 127.0.0.1:37306
Total time till object1: 0.6835460662841797 msec

Target Host = b'www.example.com'

```

- Client Side:

```

ubuntu@cs23mtech11033:~/test2$ python3 client.py
Press P if you want connection through proxy Server else press S : P
Enter Proxy Host : 127.0.0.1
Enter Proxy Port : 9697
Enter host name : www.example.com
Enter port number :80
Enter path : /
HTTP/1.1 200 OK
Age: 185821
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Sun, 05 Nov 2023 15:03:05 GMT
Etag: "3147526947+ident"
Expires: Sun, 12 Nov 2023 15:03:05 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (nyb/1D2F)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256

<!doctype html>
<html>
<head>
  <title>Example Domain</title>

  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">
    body {
      background-color: #f0f0f2;
      margin: 0;
      padding: 0;
      font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial,
sans-serif;

      margin: 0 auto;
      padding: 2em;
      background-color: #fdfdff;
      border-radius: 0.5em;
      box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
      color: #38488f;
      text-decoration: none;
    }
    @media (max-width: 700px) {
      div {
        margin: 0 auto;
        width: auto;
      }
    }
  </style>
</head>

<body>
<div>
  <h1>Example Domain</h1>
  <p>This domain is for use in illustrative examples in documents. You may use this
domain in literature without prior coordination or asking for permission.</p>
  <p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>

ubuntu@cs23mtech11033:~/test2$

```

**Code Output: For [www.iith.ac.in](https://www.iith.ac.in)**

- Proxy Side :

```
Microsoft Windows [Version 10.0.22621.2506]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\Sanket>cd documents/desktop
```

```
C:\Users\Sanket\Documents\Desktop>python p.py
Proxy server listening on 127.0.0.1:9696
Accepted connection from 127.0.0.1:54075
Total time till object1: 1.8525123596191406 msec
Target Host = b'www.iiith.ac.in'
```

- Client Side:

```
Command Prompt - python s.py
C:\Users\Sanket\Documents\Desktop>python s.py
Press P if you want connection through proxy Server else press S : P
Enter Proxy Host : 127.0.0.1
Enter Proxy Port : 9696
Enter host name : www.iiith.ac.in
Enter port number : 443
Enter path : /
HTTP/1.1 200 OK
Date: Sun, 05 Nov 2023 16:39:58 GMT
Server: Apache
Last-Modified: Sat, 04 Nov 2023 14:48:18 GMT
ETag: "dc16-60954b6262f83"
Accept-Ranges: bytes
Content-Length: 56342
Vary: Accept-Encoding
Content-Type: text/html

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link rel="icon" href="/assets/images/favicon.png">
  <!-- <title>Home | IIT Hyderabad</title> -->
  <!-- Begin Jekyll SEO tag v2.6.1 -->
  <title>Home | IIT Hyderabad</title>
  <meta name="generator" content="Jekyll v3.9.0" />
  <meta property="og:title" content="Home" />
  <meta property="og:locale" content="en" />
  <meta name="description" content="IIT Hyderabad Official Website" />
  <meta property="og:description" content="IIT Hyderabad Official Website" />
  <link rel="canonical" href="/" />
  <meta property="og:url" content="/" />
  <meta property="og:site_name" content="IIT Hyderabad" />
  <link rel="next" href="/page2" />
  <script type="application/ld+json">
    {
      "@type": "WebSite",
      "url": "/",
      "publisher": {
        "@type": "Organization",
        "logo": {
          "@type": "Image",
          "url": "/assets/images/logo.png"
        },
        "headline": "Home",
        "@context": "https://schema.org/"
      }
    }
  </script>
  <!-- End Jekyll SEO tag -->

  <link rel="stylesheet" href="/assets/css/bootstrap.min.css">
  <link href="/assets/css/font-awesome.min.css" rel="stylesheet">
  <link href="/assets/css/google-css.css" rel="stylesheet">
  <link href="/assets/css/theme.css" rel="stylesheet">
  <link href="/assets/css/faculty.css" rel="stylesheet">
  <link href="/assets/css/custom.css" rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/3.7.2/animate.min.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.min.css" integrity="sha512-NhSC1VmyruifcJ/x0yH40yVz0x50ySxdrbG46169AC05hE85yZ1s7/R5Q3z8Clwiyv2n8yAs4s526Yw4:" />
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script src="/assets/js/faculty-search.js"></script>
  <script src="/assets/js/jquery.min.js"></script>
  <link href="https://fonts.googleapis.com/css?family=Roboto&display=swap" rel="stylesheet">
  <script async src="https://www.googletagmanager.com/gtag/js?id=UA-137608318-1"></script>
  <script>
    window.dataLayer = window.dataLayer || [];
    function gtag() { dataLayer.push(arguments); }
    gtag('js', new Date());

    gtag('config', 'UA-137608318-1');
  </script>

  <style>
    #gt-nvframe{
      display: none;
    }
  </style>

</head>
```

```
Command Prompt - python s x Command Prompt Command Prompt - python p x + v
<!-- Begin Menu Navigation
===== -->
<div class="container-fluid d-none d-md-block" style="z-index: 99999;padding-bottom: 2%;">
  <span class="container-fluid" style="font-size: 0.89vw;float: right;padding-right: 0;"> English | <a href="https://translate.google.com/translate?de
pth=1&url=translate.google.co.in&sl=auto&sp=mat4&tl=hi&u=https://iith.ac.in/hindi/" target="_blank" rel="noopener noreferrer"> हिंदी</a>
  </span>
</div>

<header class="navbar navbar-toggleable-md navbar-light bg-white fixed-top mediunavigation" id="head-nav" style="padding-bottom: 1%;">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
    data-target="#navbarsWow" aria-controls="navbarsWow" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
<div class="container-fluid"></div>

<!-- Begin Logo -->
<a class="navbar-brand" href="/">
  <!-- 
  
</a>
<!-- End Logo -->

<!-- Begin Menu -->
<div class="collapse navbar-collapse" id="navbarsWow">
  <!-- Begin Menu -->
  <ul class="navbar-nav ml-auto">
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="" id="dropdown01" data-toggle="dropdown"
        aria-haspopup="true" aria-expanded="false">Academics</a>
      <div class="dropdown-menu" aria-labelledby="dropdown01" style="margin-top: unset">
        <div class="navbar-dropdown">

          <a href="/academics/index.html#admissions"
            class="dropdown-item">Admissions</a>

        </div>
      </div>
    </li>
  </ul>
</div>

===== -->
```

```
Command Prompt - python s x Command Prompt Command Prompt - python p x + v
<!-- Begin Menu Navigation
===== -->
<div class="container-fluid d-none d-md-block" style="z-index: 99999;padding-bottom: 2%;">
  <span class="container-fluid" style="font-size: 0.89vw;float: right;padding-right: 0;"> English | <a href="https://translate.google.com/translate?de
pth=1&url=translate.google.co.in&sl=auto&sp=mat4&tl=hi&u=https://iith.ac.in/hindi/" target="_blank" rel="noopener noreferrer"> हिंदी</a>
  </span>
</div>

<header class="navbar navbar-toggleable-md navbar-light bg-white fixed-top mediunavigation" id="head-nav" style="padding-bottom: 1%;">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
    data-target="#navbarsWow" aria-controls="navbarsWow" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
<div class="container-fluid"></div>

<!-- Begin Logo -->
<a class="navbar-brand" href="/">
  <!-- 
  
</a>
<!-- End Logo -->

<!-- Begin Menu -->
<div class="collapse navbar-collapse" id="navbarsWow">
  <!-- Begin Menu -->
  <ul class="navbar-nav ml-auto">
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="" id="dropdown01" data-toggle="dropdown"
        aria-haspopup="true" aria-expanded="false">Academics</a>
      <div class="dropdown-menu" aria-labelledby="dropdown01" style="margin-top: unset">
        <div class="navbar-dropdown">

          <a href="/academics/index.html#admissions"
            class="dropdown-item">Admissions</a>

        </div>
      </div>
    </li>
  </ul>
</div>

===== -->
```

```
Command Prompt - python s x Command Prompt Command Prompt - python p x + v
<!-- Begin Footer
===== -->
<footer class="footer" style="bottom: 0;">
  <div class="container">
    <div class="justify-content-around row">
      <div class="col-sm-2 d-none d-md-block">
        <div class="footer-widget">
          <h5 class="title">Academics</h5>
          <ul>
            <li><a title="Academic Institute Management System" href="https://aims.iith.ac.in/aims/"
              target="_blank">AIMS</a></li>
            <li><a title="Global Initiative of Academic Networks" href="https://www.iith.ac.in/~gian/"
              target="_blank">GIAN</a></li>
            <li><a title="Centre for Continued Education" href="http://cce.iith.ac.in/"
              target="_blank">CCE</a></li>
            <li><a title="" href="/academics/index.html">Office of Academic Affairs</a></li>
            <li><a title="" href="https://raiith.library.iith.ac.in/"
              target="_blank">RAIITH</a></li>
          </ul>
        </div>
      </div>
    </div>
  </div>

===== -->
```

## **PART-3: A Simple Web Server**

### **Code Explanation:**

- Importing Packages

```
#Importing Packages
import socket
import threading
import os
```

- Assigning port number and host to the server. And creating a socket for communication with the client.

```
#Set Host and port numbers
server_host = '127.0.0.1'
server_port = 9698

#Create socket
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((server_host, server_port))
server.listen(5)

print(f"Server is listening on {server_host}:{server_port}")
```

- Accepting Connections from the multiple clients. Created a thread to handle multiple clients.

```
while True:
    #Accept New client connection
    client_sock, address = server.accept()
    print(f"Accepted connection from {address[0]}:{address[1]}")

    #Creating thread
    multiple_clients = threading.Thread(target=handle_client, args=(client_sock,))
    multiple_clients.start()
```

- Handle Clients that is receiving request from client then split the request to fetch path for index html file.

```
#Handling Client using Threads
def handle_client(client_sock):

    #receiving client Request form proxy server or client directly
    request = client_sock.recv(1024)

    #Splitting the request
    request_lines = request.split(b'\r\n')
    print(f"Request Line : {request_lines}\n")
```

- Parsing the http request to get requested file.



```
# Parse the HTTP request to get the requested file
if len(request_lines) > 0:

    request_line = request_lines[0].decode()

    parts = request_line.split()
    print(f"Parts : {parts}\n")

    if len(parts) >= 2:
        #Removing '/'
        file_path = os.path.join('',parts[1][1:])
        print(f"File Path : {file_path}\n")

    else:
        #if no file is given it set default to index.html
        file_path = os.path.join('', 'index.html')
        print(f"File Path : {file_path}\n")
```

- Now read the file which is requested by the client. If found server send 200 OK response to the client. And along with that server send that index.html file to the client. If file not found then Server send 404 Not found message to the server.

```
try:
    #Reading Binary
    with open(file_path, 'rb') as file:
        content = file.read()

    # Status : 200 OK
    response = b'HTTP/1.1 200 OK\r\n\r\n' + content

except FileNotFoundError:

    # Status : 404 Not Found
    response = b'HTTP/1.1 404 Not Found\r\n\r\nFile Not Found'
```

- Closing connection.

```
client_sock.send(response)
client_sock.close()
```

- In server side we have a Index.html file saved. Which will be requesting by the client.

- Index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Links</title>
</head>
<body>
  <h1>Links to Other HTML Files</h1>
  <p>Click on the links below to navigate to other HTML files:</p>

  <ul>
    <li><a href="/reference1.html">Reference 1</a></li>
    <li><a href="/reference2.html">Reference 2</a></li>
    <li><a href="/reference3.html">Reference 3</a></li>
  </ul>
</body>
</html>
```

- Reference.html (Object 1)

```
<!DOCTYPE html>
<html>
<head>
  <title>Reference 1</title>
</head>
<body>
  <h1>Reference 1</h1>
  <p>This is the content of Reference 1.</p>
  
</body>
</html>
```

- Reference.html (Object 2)

```
<!DOCTYPE html>
<html>
<head>
  <title>Reference 2</title>
</head>
<body>
  <h1>Reference 2</h1>
  <p>This is the content of Reference 2.</p>
  
</body>
</html>|
```

**Code Output: For Index.html file With Proxy server**

## Server Side :

```
C:\Users\Sanket\Documents\Desktop>python servermain.py
Server is listening on 127.0.0.1:9698
Accepted connection from 127.0.0.1:53914
Request Line : [b'GET /index.html HTTP/1.1', b'Host: 127.0.0.1:9698', b'', b'']

Parts : ['GET', '/index.html', 'HTTP/1.1']

File Path : index.html

Accepted connection from 127.0.0.1:53917
Request Line : [b'GET /reference1.html HTTP/1.1', b'Host: 127.0.0.1:9698', b'', b'']

Parts : ['GET', '/reference1.html', 'HTTP/1.1']

File Path : reference1.html

Accepted connection from 127.0.0.1:53919
Request Line : [b'GET /reference2.html HTTP/1.1', b'Host: 127.0.0.1:9698', b'', b'']

Parts : ['GET', '/reference2.html', 'HTTP/1.1']

File Path : reference2.html

Accepted connection from 127.0.0.1:53921
Request Line : [b'GET /reference3.html HTTP/1.1', b'Host: 127.0.0.1:9698', b'', b'']

Parts : ['GET', '/reference3.html', 'HTTP/1.1']

File Path : reference3.html
```

## Proxy Side :

```
C:\Users\Sanket>cd documents/desktop

C:\Users\Sanket\Documents\Desktop>python p.py
Proxy server listening on 127.0.0.1:9696
Accepted connection from 127.0.0.1:53912
Target Host = b'127.0.0.1'
Total time till object1: 0.0 msec

Accepted connection from 127.0.0.1:53915
Total time till object2: 436.5355968475342 msec

Target Host = b'127.0.0.1'
Accepted connection from 127.0.0.1:53918
Total time till object3: 450.0265121459961 msec
Target Host = b'127.0.0.1'

Accepted connection from 127.0.0.1:53920
Total time till object4: 457.28516578674316 msec

Target Host = b'127.0.0.1'
```

## Client Side :

Outputs the Index file and Objects fetched from that file

```
Command Prompt - python s X Command Prompt - python p X Command Prompt
C:\Users\Sanket\Documents\Desktop>python s.py
Press P if you want connection through proxy Server else press S : P
Enter Proxy Host : 127.0.0.1
Enter Proxy Port : 9696
Enter host name : 127.0.0.1
Enter port number :9698
Enter path : /index.html
HTTP/1.1 200 OK

<!DOCTYPE html>
<html>
<head>
  <title>HTML Links</title>
</head>
<body>
  <h1>Links to Other HTML Files</h1>
  <p>Click on the links below to navigate to other HTML files:</p>

  <ul>
    <li><a href="/reference1.html">Reference 1</a></li>
    <li><a href="/reference2.html">Reference 2</a></li>
    <li><a href="/reference3.html">Reference 3</a></li>
  </ul>
</body>
</html>

['/reference1.html', '/reference2.html', '/reference3.html']
HTTP/1.1 200 OK

<!DOCTYPE html>
<html>
<head>
  <title>Reference 1</title>
</head>
<body>
  <h1>Reference 1</h1>
  <p>This is the content of Reference 1.</p>
  
</body>
</html>
```

```
.....Received Obj - 1 .....
HTTP/1.1 200 OK

<!DOCTYPE html>
<html>
<head>
  <title>Reference 2</title>
</head>
<body>
  <h1>Reference 2</h1>
  <p>This is the content of Reference 2.</p>
  
</body>
</html>

.....Received Obj - 2 .....
HTTP/1.1 200 OK

<!DOCTYPE html>
<html>
<head>
  <title>Reference 3</title>
</head>
<body>
  <h1>Reference 3</h1>
  <p>This is the content of Reference 3.</p>
  
</body>
</html>

.....Received Obj - 3 .....
```

**Code Output: For Index.html file Without Proxy server**

- **Server Side:**

```

Accepted connection from 127.0.0.1:54014
Request Line : [b'GET /index.html HTTP/1.1', b'Host: 127.0.0.1', b'', b'']

Parts : ['GET', '/index.html', 'HTTP/1.1']

File Path : index.html

Accepted connection from 127.0.0.1:54015
Request Line : [b'GET /reference1.html HTTP/1.1', b'Host: 127.0.0.1', b'', b'']

Parts : ['GET', '/reference1.html', 'HTTP/1.1']

File Path : reference1.html

Accepted connection from 127.0.0.1:54016
Request Line : [b'GET /reference2.html HTTP/1.1', b'Host: 127.0.0.1', b'', b'']

Parts : ['GET', '/reference2.html', 'HTTP/1.1']

File Path : reference2.html

Accepted connection from 127.0.0.1:54017
Request Line : [b'GET /reference3.html HTTP/1.1', b'Host: 127.0.0.1', b'', b'']

Parts : ['GET', '/reference3.html', 'HTTP/1.1']

File Path : reference3.html

```

## Client Side :

```

C:\Users\Sanket\Documents\Desktop>python s.py
Press P if you want connection through proxy Server else press S : S
Enter host name : 127.0.0.1
Enter port number : 9698
Enter path : /index.html
HTTP/1.1 200 OK

<!DOCTYPE html>
<html>
<head>
  <title>HTML Links</title>
</head>
<body>
  <h1>Links to Other HTML Files</h1>
  <p>Click on the links below to navigate to other HTML files:</p>

  <ul>
    <li><a href="/reference1.html">Reference 1</a></li>
    <li><a href="/reference2.html">Reference 2</a></li>
    <li><a href="/reference3.html">Reference 3</a></li>
  </ul>
</body>
</html>

['/reference1.html', '/reference2.html', '/reference3.html']
HTTP/1.1 200 OK

<!DOCTYPE html>
<html>
<head>
  <title>Reference 1</title>
</head>
<body>
  <h1>Reference 1</h1>
  <p>This is the content of Reference 1.</p>
  
</body>
</html>

.....Received Obj 1 .....

```

HTTP/1.1 200 OK

```
<!DOCTYPE html>
<html>
<head>
  <title>Reference 2</title>
</head>
<body>
  <h1>Reference 2</h1>
  <p>This is the content of Reference 2.</p>
  
</body>
</html>
```

.....Received Obj 2 .....

HTTP/1.1 200 OK

```
<!DOCTYPE html>
<html>
<head>
  <title>Reference 3</title>
</head>
<body>
  <h1>Reference 3</h1>
  <p>This is the content of Reference 3.</p>
  
</body>
</html>
```

.....Received Obj 3 .....

## PART 4 : Speedy Web Proxy:

### Code Explanation:

- Importing packages

```
##Importing packages
import socket
import ssl
import threading
from urllib.parse import urlparse
import time
```

- Assigning host and port to speedy proxy and create Socket.

```
#assignning Host and port number to the speedy proxy
proxy_host = '127.0.0.1'
proxy_port = 9696

#Created Sockets for Communication
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((proxy_host, proxy_port))
server.listen(5)

print(f"Proxy server listening on {proxy_host}:{proxy_port}")
```

- Accepting client requests and start timer and use threading for Multiple client requests.

```
while True:

    #Accepting Client requests
    client_sock, address = server.accept()
    i = i+1
    if (flag == 0):
        #store Start time
        start_time = time.time()
        flag = 1

    print(f"Accepted connection from {address[0]}:{address[1]}")
    multiple_clients = threading.Thread(target=handle_client, args=(client_sock,))
    multiple_clients.start()
```

- Calculate total time

```
#store End time
end_time = time.time()

#Calculate Total time required
total_time = (end_time - start_time)*1000
print(f"Total time till object{i}: {total_time} msec\n")
```

- Handle Client, Receive response from client. Then create a socket and make a connection.

```
def handle_client(client_sock):
    try:
        #receiving Client request
        request = client_sock.recv(4096)

        if request:

            target_host, target_port = extract_host(request)
            print(f"Target Host = {target_host}")

            if target_host:
                # Create a socket
                server_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

                # Connect to Target Host and Target Port
                server_sock.connect((target_host, target_port))
```

- Now it will Extract Host.

```
def extract_host(request):
    # Extract the Host header from the HTTP request
    # Extract host and port number from http request
    for line in request.split(b'\n'):
        if line.startswith(b'Host:'):
            host_line = line.split(b' ')[1].strip()
            host_parts = host_line.split(b':')

            if len(host_parts) == 2:
                host, port = host_parts
                return host, int(port)
            else:
                # using 80 as a default port if not specified
                host = host_parts[0]
                return host, 80
```

- proxy Send request and receives the response.

```
server_sock.send(request)

# Receive the server's response
while True:
    server_response = server_sock.recv(4096)

    if len(server_response) == 0: # No response
        break

    # Check if this is the base HTML response
    if b'Content-Type: text/html' in server_response:
        # Parse HTML for objects and fetch them
        server_response = parse_and_fetch_objects(server_response, target_host, target_port)

    client_sock.send(server_response)
```



- Now it will go to the parse and fetch function.

```
#Parsing and fetching objects from html response
def parse_and_fetch_objects(html_response, target_host, target_port):

    # Parse the HTML response to find object URLs and fetch them
    html_response = html_response.decode('utf-8', errors='ignore')
    lines = html_response.split('\n')
    new_html = []
    object_urls = set()

    for line in lines:
        if line.strip().startswith('<script') or line.strip().startswith('<img') or line.strip().startswith('<link'):
            url = get_object_url(line)
            if url:
                object_urls.add(url)

    # Fetch objects in parallel using threading
    object_responses = fetch_objects_in_parallel(object_urls, target_host, target_port)
```

- After parsing and fetching now it will fetch parsed object in parallel tcp connections.

```
#Parallel Object Fetching (Parallel Connection)
def fetch_objects_in_parallel(object_urls, target_host, target_port):

    #fetching objects Parallely
    object_responses = {}
    threads = []

    for url in object_urls:
        t = threading.Thread(target=fetch_object, args=(url, target_host, target_port, object_responses))
        t.start()
        threads.append(t)

    for t in threads:
        t.join()

    return object_responses
```

- In fetch objects in parallel it will call fetch object.

```
def fetch_object(url, target_host, target_port, object_responses):

    # Fetch an object from the target server
    try:
        #created socket for fetching objects
        object_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        parsed_url = urlparse(url)
        object_socket.connect((target_host, target_port))

        if parsed_url.scheme == 'https':
            object_socket = ssl.wrap_socket(object_socket, server_side=False)

        object_request = f"GET {parsed_url.path} HTTP/1.1\r\nHost: {target_host}\r\n\r\n".encode('utf-8')
        object_socket.send(object_request)

        response = b""
        while True:
            chunk = object_socket.recv(4096)
            if not chunk:
                break
            response += chunk

        object_responses[url] = response
        object_socket.close()
```

## References :

implementation, please let me know, and I'd be happy to help. However, if you're looking for guidance on building a multi-threaded web server from scratch, here are some steps to get you started:

1. Import necessary modules:

You'll need to import Python modules for socket programming and threading. In your code, it seems you've already imported the socket module.

2. Create a socket for listening to incoming connections:

In your main function, create a socket that listens for incoming connections from clients.

You can use `socket.socket(socket.AF_INET, socket.SOCK_STREAM)` to create a socket for accepting connections.

3. Bind the socket to a specific host and port:

Use the `bind` method to bind the socket to a specific host and port where your web server will listen for incoming connections.

4. Listen for incoming connections:

Use the `listen` method to start listening for incoming connections. You can specify the maximum number of queued connections as an argument.

5. Accept incoming connections in a loop:

Use a loop to continuously accept incoming client connections. When a connection request is accepted, create a new thread to handle that connection. You can use the `threading` module to create and start a new thread for each connection.

Regenerate

6. Handle HTTP requests in separate threads:

In the thread's function, you can implement the logic to handle HTTP requests from clients.

This involves parsing the HTTP request, fetching the requested file, creating an HTTP response, and sending it back to the client. Be sure to handle the case where the requested file is not found (return a "404 Not Found" response).

7. Close the client connection:

After sending the HTTP response to the client, make sure to close the client's socket connection.

8. Test your server:

Run your server and use your web client to send HTTP requests to it, just like you described in your question. You should be able to access the HTML files placed in the server directory and receive proper responses, including "404 Not Found" for missing files.

Note: In a production environment, you might want to add error handling, logging, and potentially implement security measures.

I hope this helps you get started with building your multi-threaded web server. If you have any specific questions or need help with a particular aspect of your implementation, please feel free to ask.

Regenerate

**Code Output: For [gaia.cs.umass.edu](http://gaia.cs.umass.edu)**

## Using Simple Proxy Server :

```
ubuntu@cs23mtech11033:~/test2$ python3 proxy.py
Proxy server listening on 127.0.0.1:9697
Accepted connection from 127.0.0.1:53622
Target Host = b'gaia.cs.umass.edu'
Total time till object1: 0.9005069732666016 msec

Accepted connection from 127.0.0.1:53638
Total time till object2: 5495.803594589233 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:37614
Total time till object3: 10982.885122299194 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:37618
Total time till object4: 16472.347021102905 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:49416
Total time till object5: 21961.028814315796 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:42208
Total time till object6: 27532.850980758667 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:42214
Total time till object7: 33026.856660842896 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:53740
Total time till object8: 38520.092487335205 msec

Target Host = b'gaia.cs.umass.edu'
█
```

## Code Output: For gaia.cs.umass.edu

### Using Fast/Speedy Proxy Server :

```
ubuntu@cs23mtech11033:~/test2$ python3 ExtendedProxy.py
Proxy server listening on 127.0.0.1:9696
Accepted connection from 127.0.0.1:57796
Target Host = b'gaia.cs.umass.edu'
Total time till object1: 0.9458065032958984 msec

Accepted connection from 127.0.0.1:57810
Total time till object2: 5498.652458190918 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:59466
Total time till object3: 10992.324829101562 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:59468
Total time till object4: 16480.80277442932 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:57596
Total time till object5: 21968.74475479126 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:57608
Total time till object6: 27456.384420394897 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:57498
Total time till object7: 32944.32783126831 msec

Target Host = b'gaia.cs.umass.edu'
Accepted connection from 127.0.0.1:54456
Total time till object8: 38435.45341491699 msec

Target Host = b'gaia.cs.umass.edu'
```

Here By comparing time required to fetch objects through Simple proxy and Speedy proxy. We can say that the **speedy proxy** required **less time** to fetch objects than simple proxy.