



Department of Electrical and Computer Engineering

ENCS3320-Computer Networks

Project#1 due 30/5/2023

- 1- This is a group project, so you are allowed to work in groups of max 3 students
- 2- Do not use libraries to implement the project. Use socket programming
- 3- **Important: Each screenshot should include the date and time of your computer.**

You have to submit

- 1- A report in pdf format (only pdf format) on moodle (itc.birzeit.edu) that contains **Screenshots** with **detailed explanation, codes, runs, etc.**
- 2- The code with comments (include the code in the pdf file and as text file .py or .java or .c as well)
- 3- You are allowed to send compressed file (e.g., .zip). But you have to send the report as pdf file separately.

Part1:

- 1- In your own words, what are ping, tracer, nslookup, and telnet (write one sentence for each one)
- 2- Make sure that your computer is connected to the internet and then run the following commands:
 - 1- Ping a device in the same network, e.g. from a laptop to a smartphone
 - 2- ping www.harvard.edu
 - 3- tracer www.harvard.edu
 - 4- nslookup www.harvard.edu

Provide a screenshot of the runs and brief explanation of the output.

Part2:

Using socket programming, implement UDP client and server applications in go, python, java or C. The server should listen on port 8855. The client sends broadcast messages every 2 seconds. The message should contain the student's name. The server lists the last received message from a client. If 3 clients sent messages, the server should display 4 lines something like

Server **First name Last name**

- 1- received message from **First name Last name** at **Time**
- 2- received message from **First name Last name** at **Time**
- 3- received message from **First name Last name** at **Time**

where **First name Last name** is the sender first and last names. **Time** is the last time a message received from that sender.

You can use the IP address of the sender to distinguish between different senders.

The broadcast address of a network with subnet mask 255.255.255.0 is the similar to the IP address of a host in that network with last octet **255**. For instance, if the IP address is 192.168.1.12 and subnet mask is 255.255.255.0, then the broadcast address is 192.168.1.**255**.

If the subnet mask 255.255.0.0 then the last 2 octets of broadcast IP are **255.255**
Each student should run a client or server (group of 3 students should contain at least 2 clients and one server)

For each run, provide screenshots of the run and provide a brief description.

Make sure that the computers are within the same subnet. For example, the IP of the first computer is 192.168.1.10, subnet mask 255.255.255.0 and the IP of the second computer is 192.168.1.11 and the subnet mask 255.255.255.0. the IP of the third computer is 192.168.1.12 and the subnet mask 255.255.255.0.

Part3:

Using socket programming, implement a simple but a complete web server in go, python, java or C that is listening on port 9977. The user types in the browser something like <http://localhost:9977/ar> or <http://localhost:9977/en>

The program should check

- 1- if the request is **/ or /index.html or /main_en.html or /en (for example localhost:9977/ or localhost:9977/en)** then the server should send **main_en.html** file with Content-Type: text/html.

The **main_en.html** file should contain

HTML webpage that contains

- a. "ENCS3320-My Tiny Webserver" in the title
 - b. "Welcome to our course **Computer Networks, This is a tiny webserver**" (part of the phrase is in **Blue**)
 - c. Group members names and IDs
 - d. Some information about the group members. For instance, projects you have done during different course (programming, electrical, math, etc), skills, hobbies, etc.
 - e. Use CSS to make the page looks nice
 - f. Divide the page in different boxes and put student's information in the different boxes
 - g. Include CSS as a separate file
 - h. The page should contain at least An image with extension.jpg and an image with extension .png
 - i. A link to a local html file (an html file)
 - j. a link to https://www.w3schools.com/python/gloss_python_multi_line_strings.asp
- 2- If the request is **/ar** then the server response with **main_ar.html** which is an Arabic version of main_en.html

- 3- if the request is an **.html** file then the server should send the requested html file with Content-Type: text/html. You can use any html file.
- 4-
- 5- if the request is a **.css** file then the server should send the requested css file with Content-Type: text/css. You can use any CSS file
- 6-
- 7- if the request is a **.png** then the server should send the png image with Content-Type: image/png. You can use any image.
- 8- if the request is a **.jpg** then the server should send the jpg image with Content-Type: image/jpeg. You can use any image.
- 9- Use the status code **307 Temporary Redirect** to redirect the following
 - a. If the request is **/yt** then redirect to youtube website
 - b. If the request is **/so** then redirect to stackoverflow.com website
 - c. If the request is **/rt** then redirect to ritaj website

- 10- If the request is wrong or the file doesn't exist the server should return a simple HTML webpage that contains (Content-Type: text/html)
 - 1- "HTTP/1.1 404 Not Found" in the response status
 - 2- "Error 404" in the title
 - 3- "**The file is not found**" in the body in **red**
 - 4- Your names and IDs in **Bold**
 - 5- The IP and port number of the client

- 11- The program should print the **HTTP requests** on the **terminal window** (command line window).

Provide **screenshots** of the browser with brief descriptions to show that your project works as expected. (**/main_en.html /imagename.png, /yt, etc.**) . Test the project from a browser on the **same computer** and from **a different computer or phone**.

Provide also a **screenshot** of the **HTTP request** printed on the command line.

Hint: Have a look on HTTP response in Listing 1 and the HTML code In Listing 2. You may use the minimal header and HTML code. Have a look also on rfc2616 (<https://tools.ietf.org/html/rfc2616>)

```
HTTP/1.1 200 OK
Connection: close
Date: Fri, 03 Mar 2017 06:19:37 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.1e-fips PHP/5.4.16
Last-Modified: Fri, 03 Mar 2017 05:28:07 GMT
Content-Length: 6821
```

*Content-Type: text/html
data data data data data ...*

Listing 1: HTTP Response

```
<!DOCTYPE html>  
<html>  
<head><title >XYZ Company INC.</ title ></head>  
<body><h1>Welcome <b>XYZ</b> Company</h1>  
<br>  
We are so happy that you have chosen to visit our website.  
</body>  
</html>
```

Listing 2: Simple HTML Code