Department of Electrical and Computer Engineering
Computer Networks (ENCS3320)
Project#1: Socket Programming, due 14/08/2024

Objectives

- To be familiar with some set of commands in term of the networking course.
- To be familiar with creating TCP and UDP sockets.
- To be familiar with the basics of the HTML and web server.
- To work in teams.

Tasks:

Task1: Commands & Wireshark

1. In your words, what are **ping**, **tracert**, **nslookup**, and **telnet**?
2. Make sure that your computer is connected to the internet and then run the following commands:
   a. **ping** a device in the **same** network, e.g. from a laptop to a smartphone.
   b. **ping** *www.ox.ac.uk*.
   c. From the ping results, specify the location of the server from where you got the response? Explain your answer briefly.
   d. **tracert** *www.ox.ac.uk*.
   e. **nslookup** *www.ox.ac.uk*.
   f. **telnet** *www.ox.ac.uk*.
3. Give some details about autonomous system (AS) number, number of IPs, prefixes, peers, name of Tier1-ISP of *www.ox.ac.uk*.
   - **Hint**: you can use online tool (*www.bgpview.io*) to answer this point.
4. Use **wireshark** (free tool and available online) to capture some DNS messages.

Give **screenshots** (*with time and date of your machine*) for all **runs** and **explanation** of the outputs.

Task2: Socket Programming (TCP and UDP)

1. Using socket programming, write simple TCP client server python applications (*in go, python, java or C*) to send input data from client to server, replace all the vowels (aeiou/AEIOU) with '#' and return the string to client and print it. You need to choose the port number based on the std.ID of one of the students in the team (e.g. 120**1515** → port is **1515**) for this communication.

2. Using socket programming, implement UDP client and server applications (*in go, python, java or C*). The server (peer) should listen on a port number; this number should be selected based on the std.ID of one of the students in the team (e.g. 120**1515** → port is **1515**).
   All peers (clients and server) can send and receive messages. In other words, a message sent by a peer will be received by another peer called server at this moment, others can also send to the same peer (server) after each other. The message should include peer (client) and its number as well as any message (e.g. "Hello"). The peer (server) lists the last received message from a peer (client) based on its communication with all peers. You can have this style of printing for server in the communication between peer (server) and peer (client 1):

Message from **client 1**:
Enter your message to **client 1**:
.
.
.
Message from **client 3**:
Enter your message to **client 3**:

The server will show all communications between all peers, while each client needs to show only the communication done with the server. You need to test all other peers in the same connection.

Give **screenshots** (*with time and date of your machine*) for all **runs**, **codes**, and **explanation** of the outputs for (**1**) and (**2**).

Task3: Web Server

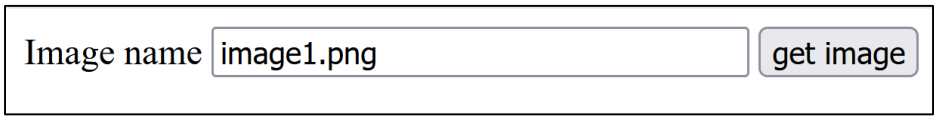
Using socket programming, implement a simple but a complete web server (*in go, python, java or C*) that is listening on a port number; this number should be selected based on the std.ID of one of the students in the team (e.g. 120**1515** → port is **1515**). Make the code as general as possible. The user types in the browser something like http://localhost:1515/ar or http://localhost:1515/en.

The program should check:

1- If the request is / or /index.html or /main_en.html or /en (for example localhost:1515/ or localhost:1515/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 First Webserver" in the title.
   b. "Welcome to **Computer Networks, ENCS3320-Webserver**" **(**part of the phrase is in **RED).**
   c. Group members' names and IDs.
   d. Some information about the group members. For example, 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 (mySite**STDID**.html).
   j. A link to *https://www.birzeit.edu/*.

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 **.html** file, then the server should send the requested html file with Content-Type: text/html. You can use any html file. Make it general (not only for specific filename).
4- If the request is **.css** file, then the server should send the requested css file with Content-Type: text/css. You can use any CSS file. Make it general (not only for specific filename).
5- If the request is **.png**, then the server should send the png image with Content-Type: image/png. You can use any image. Make it general (not only for specific filename).

6- If the request is **.jpg**, then the server should send the jpg image with Content-Type: image/jpeg. You can use any image. Make it general (not only for specific filename).
7- Store some images in a folder.
8- Use mySiteSTDID.html to get image by typing the name of the image in a box
For instance:



9- Use the status code **307 Temporary Redirect** to redirect the following;
   a. If the request is **/so** then redirect to stackoverflow.com website.
   b. If the request is **/itc** then redirect to itc.birzeit.edu 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)
   - "HTTP/1.1 404 Not Found" in the response status
   - "Error 404" in the title
   - "The file is not found" in the body in **BLUE**
   - Your names and IDs in **Bold**
   - The IP and port number of the client
11- The program should print the HTTP requests on the terminal window (command line window).

Give screenshots (*with time and date of the machine*) of the browser with brief descriptions to show that your project works as expected for all requested links **(/main_en.html /imagename.png, /itc, etc.)**. Test the project from a browser on the **same computer** and from **a different computer or phone**. Give also a screenshot (*with time and date of the machine*) of the HTTP request printed on the command line.

Project Report:

The report must contain parts highlighting the following points:

I. Cover Page
- Contains the university logo, department, course name and number, project title, your name (and id) for all team members and their section, and date.

II. Theory and Procedure Part
- Describe the theory for each required point and try to arrange the ideas based on the given titles for each part.
- List the components/ideas required to solve each part, and the reason/comment why each component/idea is used. Provide diagrams/flowcharts for the solutions with full description on the figures. Don't forget to add captions to all figures and call them in the description.
- Add some citations for all ideas and theories, don't forget to add references at the end of the report.

III. Results and Discussions Part
- Do not use libraries to implement the project. Use socket programming (only socket library is allowed).
- For each required point in the document, you need to show the output and discuss it. Don't list only screenshots/codes without any details. Also, don't add figures without captions or without calling them within the paragraphs.

- If the point needs to be solved by a code, you need to add the code as snapshot within this part as well as you need to add the code itself as .txt file in the submitted .zip file.
- Provide snapshots showing the results, each snapshot must contain the time and the date of your machine.

IV. Alternatives Solutions, Issues and Limitations Part

- Describe the alternative solutions for any question in the project rather than the one required in the document.
- Show the limitations, issues, and challenges you have faced during the project with your team or alone. You need also to show the not working parts (if any).

V. Teamwork Part

- Three students can form a team at max, students could be from any section. Write the names of all team members on the project report cover page.
- Team members need to coordinate their work among themselves, so everyone will participate in design, implementation, simulation, testing, and report.
- Show the work done by each member in the team using a chart (i.e. bar, histogram, pie), and list the parts of each one in the team.

NOTE: Don't forget to add the appropriate numbering style, table of contents, list of figures, list of tables, references, and appendices.

Project Submissions Format and Deadlines:

A. You need to submit the required documents through **ITC** before end of Wednesday Aug.14, 2024.
B. One submission per team is enough and the last submission will be only considered for the project.
C. Files to be submitted through **ITC** session as follows:
   i. A detailed report in pdf format (only **.pdf** format) for the project based on previous parts. The name of this pdf should be linked with your group number assigned in the excel sheet (e.g. **G1.pdf**). Deductions will be applied on other extensions or names.
   ii. A compressed file (**.zip**) and name it by your group number (**G1.zip**), this file contains:
      1. Codes/commands for **Part1**, **Part2_1**, and **Part2_2** in **.txt** files. You need to name the file by **G1_Part1.txt** for first part and so on for **Part2_1** and **Part2_2**.
      2. Folder contains all related files of Part3 like (.py, .css, and .html). This folder should be named by **G1_Part3** based on the group number.

Grading Criteria

This is some sort of project competition. So one factor of the evaluation depends on how a project distinguishes itself. The evaluation factors that may distinguish your project:

- The more properly working features the better (i.e. a project of three well designed and implemented features is better than four poorly designed or implemented features).
- How well the parts are integrated with each other to serve the project goals.
- The level of understanding the details of implementation.
- How difficult it is to interface and use the parts (relatively).

The following table summarizes the grading criteria and submission deadlines of the course project.

Project Component	**Percentage**	**Submission Deadline**
Project Report	75%	Wednesday, Aug. 14, 2024
Discussion/Question in Final Exam	25%	Final Exam

Generally, just by following the guidelines presented in this document, you should get a good score. However, failing to stick to these guidelines may result in a reduction proportional in magnitude to the deviation.

Good luck, *ENCS3320 Instructor*