**`Distributed Systems Lab**

**Assignment 1:**

**Title: "Introduction to Inter-Process Communication Using Sockets"**

**Objective:**

The goal of this assignment is to get introduced to inter-process communication (IPC) in a distributed environment using sockets. You will implement a simple client-server application where multiple clients can communicate with a server over a network.

**Assignment Details:**

1. **Task:**
   * Create a client-server application using TCP/IP sockets.
   * The server will listen on a specific port and handle multiple clients concurrently.
   * Each client will connect to the server and send a message (e.g., a simple string).
   * The server will respond to the client with a confirmation message, such as "Message received: [Client Message]".
2. **Requirements:**
   * **Server:**
     + Implement a multi-threaded or asynchronous server that can handle multiple client connections simultaneously.
     + The server should log all incoming messages with timestamps.
   * **Client:**
     + Implement a client that connects to the server and sends a user-input message.
     + The client should display the server's response and then terminate.
   * **Protocol:**
     + Use a simple request-response protocol over TCP.
     + Ensure that the server can handle and respond to clients in a fault-tolerant manner.
3. **Submission:**
   * Source code for both the client and server applications.
   * A brief report explaining the design and implementation, including any challenges faced and how they were overcome.
4. **Tools & Languages:**
   * Students may use any programming language that supports socket programming (e.g., Python, Java, C++).

**Server Code**

import socket

def start\_server():

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) [Create a socket object]

host = socket.gethostname() [ Get local machine name]

port = 9999

server\_socket.bind((host, port)) [Bind to the port]

server\_socket.listen(5) [Queue up to 5 requests]

print(f"Server started! Listening on {host}:{port}")

while True:

client\_socket, addr = server\_socket.accept() [Establish a connection]

print(f"Got a connection from {addr}")

message = 'Thank you for connecting' + "\r\n" [Send a thank you message to the client]

client\_socket.send(message.encode('ascii'))

client\_socket.close() [Close the connection]

if \_\_name\_\_ == "\_\_main\_\_":

start\_server()

**Client Code**

import socket

def start\_client():

client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

[Create a socket object]

host = socket.gethostname() [Get local machine name]

port = 9999

client\_socket.connect((host, port)) [Connection to hostname on the port]

message = client\_socket.recv(1024) [Receive no more than 1024 bytes]

print(message.decode('ascii'))

client\_socket.close() [Close the connection]

if \_\_name\_\_ == "\_\_main\_\_":

start\_client()