Practical Work 1: TCP File Transfer

Tran Duc Manh - 22BI13274

29/11/2024

Goal

The objective of this project is to implement a file transfer system over TCP/IP using socket programming. The system includes:

- A server that receives the file and saves it.
- A client that sends the file to the server.
- Communication via TCP sockets.

Protocol Design

The protocol utilizes TCP (Transmission Control Protocol), which ensures reliable and ordered communication. The flow is as follows:

- The server initializes a socket, binds to a port, and listens for incoming connections.
- The client connects to the server and sends file data in chunks.
- The server receives the data and writes it to a file on disk.
- Both client and server close the connection upon completion.

Figure 1: TCP File Transfer Protocol Flow

System Organization

The system consists of two components:

- **Server**: A Python script that listens for connections and receives files.
- **Client**: A Python script that reads a file and sends it to the server.

Figure 2: System Organization Diagram

Implementation

The implementation is done in Python, with separate scripts for the server and client.

Server Code

Below is the server implementation:

Listing 1: Server Code: tcp$_f$*ile*$_s$*erver.py*

```
import socket
def start_server(host='127.0.0.1', port=65432, output_file='received_f
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind((host, port))
    server_socket.listen(1)
    print(f"Server listening on {host}:{port}...")
    conn, addr = server_socket.accept()
    print(f"Connected by {addr}")
    with open(output_file, 'wb') as file:
        while True:
            data = conn.recv(1024)
            if not data:  # Connection closed
                break
            file.write(data)
    print(f"File received and saved as {output_file}.")
    conn.close()
    server_socket.close()
if __name__ == "__main__":
    start_server()
```

Client Code

Below is the client implementation:

Listing 2: Client Code: tcp$_{f}ile_{c}lient.py$

```
import socket
def send_file(file_path, host='127.0.0.1', port=65432):
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect((host, port))
    with open(file_path, 'rb') as file:
        while (chunk := file.read(1024)):
            client_socket.sendall(chunk)
    print(f"File␣{file_path}␣has␣been␣sent␣to␣the␣server.")
    client_socket.close()
if __name__ == "__main__":
    send_file('file_to_send.txt')  # Replace with the actual file you
```

Conclusion

This project successfully demonstrated file transfer over TCP using Python. It showcased the reliable nature of TCP and provided practical experience in socket programming.