

Computer Network: Socket UDP Client-Server Application

Simple chat application using UDP (User Datagram Protocol) for communication between a client and a server. UDP is a connectionless protocol, making it suitable for applications where a small amount of data needs to be transmitted quickly.

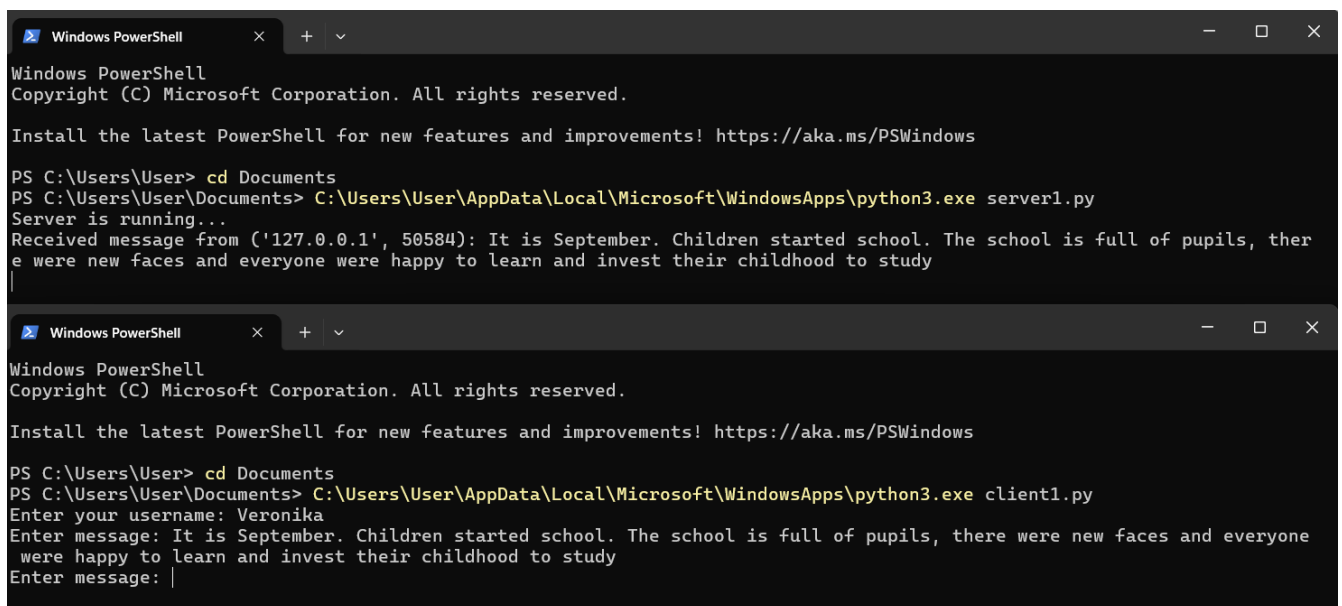
Components:

1. Server:

- Create a UDP socket and bind it to a specific port.
- Listen for incoming messages from clients.
- Upon receiving a message, process it and send a response back to the client.
- Handle multiple clients concurrently using threading or asynchronous programming.

2. Client:

- Create a UDP socket.
- Allow the user to input messages to send to the server.
- Send messages to the server.
- Receive responses from the server and display them to the user.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\User> cd Documents
PS C:\Users\User\Documents> C:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.exe server1.py
Server is running...
Received message from ('127.0.0.1', 50584): It is September. Children started school. The school is full of pupils, there were new faces and everyone were happy to learn and invest their childhood to study

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\User> cd Documents
PS C:\Users\User\Documents> C:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.exe client1.py
Enter your username: Veronika
Enter message: It is September. Children started school. The school is full of pupils, there were new faces and everyone were happy to learn and invest their childhood to study
Enter message: |
```

Implemented this UDP connection in Python:

Client code:

```
import socket
```

```
def main():
```

```
    server_ip = '127.0.0.1'
```

```
    server_port = 12000
```

```
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```

username = input('Enter your username: ')

while True:
    message = input('Enter message: ')
    client_socket.sendto(message.encode(), (server_ip, server_port))

if __name__ == "__main__":
    main()

```

Server code:

```

import socket

def main():
    server_ip = '127.0.0.1'
    server_port = 12000

    server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    server_socket.bind((server_ip, server_port))

    print('Server is running...')

    while True:
        message, client_address = server_socket.recvfrom(2048)
        message = message.decode()
        print(f'Received message from {client_address}: {message}')

if __name__ == "__main__":
    main()

```