A Project Report

On

Socket programming using Python

Submitted to

In partial fulfillment of the requirements for the award of the degree

Of

Bachelor of Science

In

Computer Science and Information Technology

[Fifth semester]

Submitted by :

Padam Raj Rijal

Under the guidance of

Mr. Yogesh Aryal

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

MADAN BHANDARI MEMORIAL COLLEGE

BINAYAKNAGAR, NEW BANESHWOR, KATHMANDU

Table of contents

CHAPTER-1

ACKNOWLEDGEMENT

This project has been completed by **Padam Raj Rijal**. I am very thankful to our teacher **Mr. Yogesh Aryal** who has guided me in completing this project. I am thankful to Department of Computer Science, college administration are equally helpful to me in making this project.

I would also like to thank **Mr. Mandeep Kharel** who helped clear any doubt what so ever I had to encounter while making this project.

It is a matter of great pleasure for me to express our deep sense of gratitude and respect to **Mr. Harendra Bist** and **Mr. Yogesh Aryal** who were there on every step to guide me in and helped me make the Project better.

CHAPTER-2

INTRODUCTION

This report presents a details overview in developing a client-server based chat application using socket programming. The application is developed using C programing. The primary objective of this report is to present the principles behind socket programming and the libraries available for socket programming applications in python.

Chatting is a method of using technology to bring people and ideas together despite of the geographical barriers. The technology has been available for years but the acceptance it was quite recent. This project is an example of a multiple client chat server. It is made up of 2 applications the client application, which runs on the user's pc and server application which runs on any pc on network. To start chatting client should get connected to server. We will focus on TCP and UDP
socket connections which are a fundamental part of socket programming.

CHAPTER-3

FLOWCHART

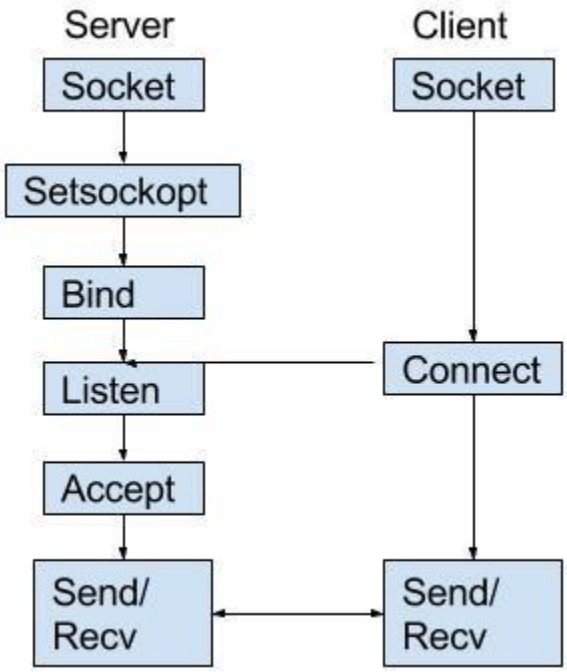

Fig1: Flowchart of Server-client system

CHAPTER-4

PROCEDURE IN CLIENT-SERVER COMMUNICATION

- **Socket:** Create a new communication
- **Bind:** Attach a local address to a socket
- **Listen:** Announce willingness to accept connections
- **Accept:** Block caller until a connection request arrives
- **Connect:** Actively attempt to establish a connection
- **Send:** Send some data over connection
- **Receive:** Receive some data over connection
- **Close:** Release the connection

CHAPTER-5

SYSTEM SPECIFICATION

Hardware requirements

In hardware requirement we require all those components which will provide us the platform for the development of the project. The minimum hardware required for the development of this project is as follow :-

Ram minimum 128 MB
Hard disk—minimum 5 GB
Processor- Pentium 3

These all are the minimum hardware requirement required for our project. I want to make this project to be used in any. Type of computer therefore I have taken minimum configuration to a large extent.128 MB ram is used so that we can execute this project in a least possible RAM. Hard disk is used because project takes less space to be executed or stored. Therefore, minimum hard disk is used. Others enhancements are according to the needs.

Software requirements

Software can be defined as the programs which runs on our computer. It is very important to run software to run computer. Various software's are needed in this project for its development.

Operating system: Windows 10, python 3.7 and notepad++

CHAPTER-6

DEVELOPMENT OF SOFTWARE

Socket Overview

A socket is an object that represents a low-level access point to the IP stack. This socket can be opened or closed or one of a set number of intermediate states. A socket can send and receive data down disconnection. Data is generally sent in blocks of few kilobytes at a time for efficiency; each of these blocks are called *a packet*. All packets that travel on the internet must use the Internet Protocol. This means that the source IP address, destination address must be included in the packet. Most packets also contain a port number. A port is simply a number between 1 and 65,535 that is used to differentiate higher protocols. Ports are important when it comes to programming your own network applications because no two applications can use the same port.

A socket is simply an endpoint for communications between the machines. The Socket class can be used to create a socket. Socket Programming is used for communication between machines using
a Transfer Control Protocol (TCP). It can be connectionless or connection-oriented. Server Socket and Socket classes are used for connection-oriented socket programming. After creating a connection, the server develops a socket object on its end of the connection. The server and client now start communicating by writing to and reading from the socket. The client needs to know two basic information, which are: Port number& IP address of server.

If everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to the same local port and also has its remote endpoint set to the address and port of the client. It needs a new socket so that it can continue to listen to the original socket for connection requests while tending to the needs of the connected client. On the client side, if the connection is accepted, a socket is successfully created and the client can use the socket to communicate with the server. The client and server can now communicate by writing to or reading from their sockets

Project Scope

This project can be mainly divided into two modules:

1. Server

2. Client

This project is mainly depended on client/server model. The client requests the server and server responses by granting the clients request. The proposed system should provide both of the above features along with the followed ones:

Server

A server application normally listens to a specific port waiting for connection requests from a client. When a connection request arrives, the client and the server establish a dedicated connection over which they can communicate. During the connection process, the client is assigned a local port number, and binds a *socket* to it. The client talks to the server by writing to the socket and gets information from the server by reading from it. Similarly, the server gets a new local port number (it needs a new port number so that it can continue to listen for connection requests on the original port). The server also binds a socket to its local port and communicates with the client by reading from and writing to it. The client and the server must agree on a protocol that is, they must agree on the language of the information transferred back and forth through the socket. Normally, a server runs on a specific computer and has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request.

Development of server

The code involved in the development of server are as follows:-

```
import socket
LOCALHOST = "127.0.0.1"
PORT = 8080
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((LOCALHOST, PORT))
server.listen(1)
print("Server started")
print("Waiting for client request..")
clientConnection,clientAddress = server.accept()
print("Connected client :" , clientAddress)
msg = ''
while True:
  in_data = clientConnection.recv(1024)
  msg = in_data.decode()
  if msg=='bye':
    break
  print("From Client :" , msg)
  out_data = input()
  clientConnection.send(bytes(out_data,'UTF-8'))
print("Client disconnected....")
clientConnection.close()
```

Client

On the client site the client knows the hostname of the machine on which the server is running and the port number on which the server is listening. To make a connection request, the client tries to rendezvous with the server on the server's machine and port. The client also needs to identify itself to the server so it binds to a local port number that it will use during this connection. This is usually assigned by the system.

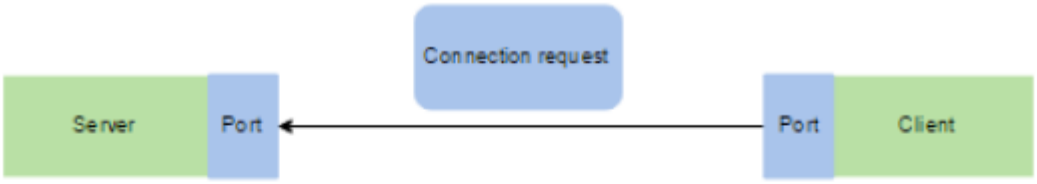

Fig2: Client reguesting for connection to server

Fig3: Client requesting for connection to server

Development of client

The code involved in the development of client are as follows:-

```
import socket
SERVER = "127.0.0.1"
PORT = 8080
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect((SERVER, PORT))
client.sendall(bytes("This is from Client",'UTF-8'))
while True:
  in_data =  client.recv(1024)
  print("From Server :" ,in_data.decode())
  out_data = input()
  client.sendall(bytes(out_data,'UTF-8'))
  if out_data=='bye':
    break
client.close()
```

CHAPTER-7

HOW TO RUN SERVER.PY AND CLIENT.PY FILES?

Create Python Server Socket Program (Server.py) and Python **Client Socket Program** (client.py) in two separate files. When you finish coding, first you have to start Python Server Socket Program from **DOS prompt** (console) , then you will get a message " Server Started..." and "Waiting for client request.." in your DOS screen, where the server program is running .

Next step is to start Python **Client Socket Program** from DOS prompt (console) in the same computer or other computers on the same network . When you start the client program , it will establish a connection to the Server and send message ("This is from Client") from client side. After receiving message from client side, the server send message to the client "Successfully Connected to Server!!". That's now you can see you client program and server program communicate each other.

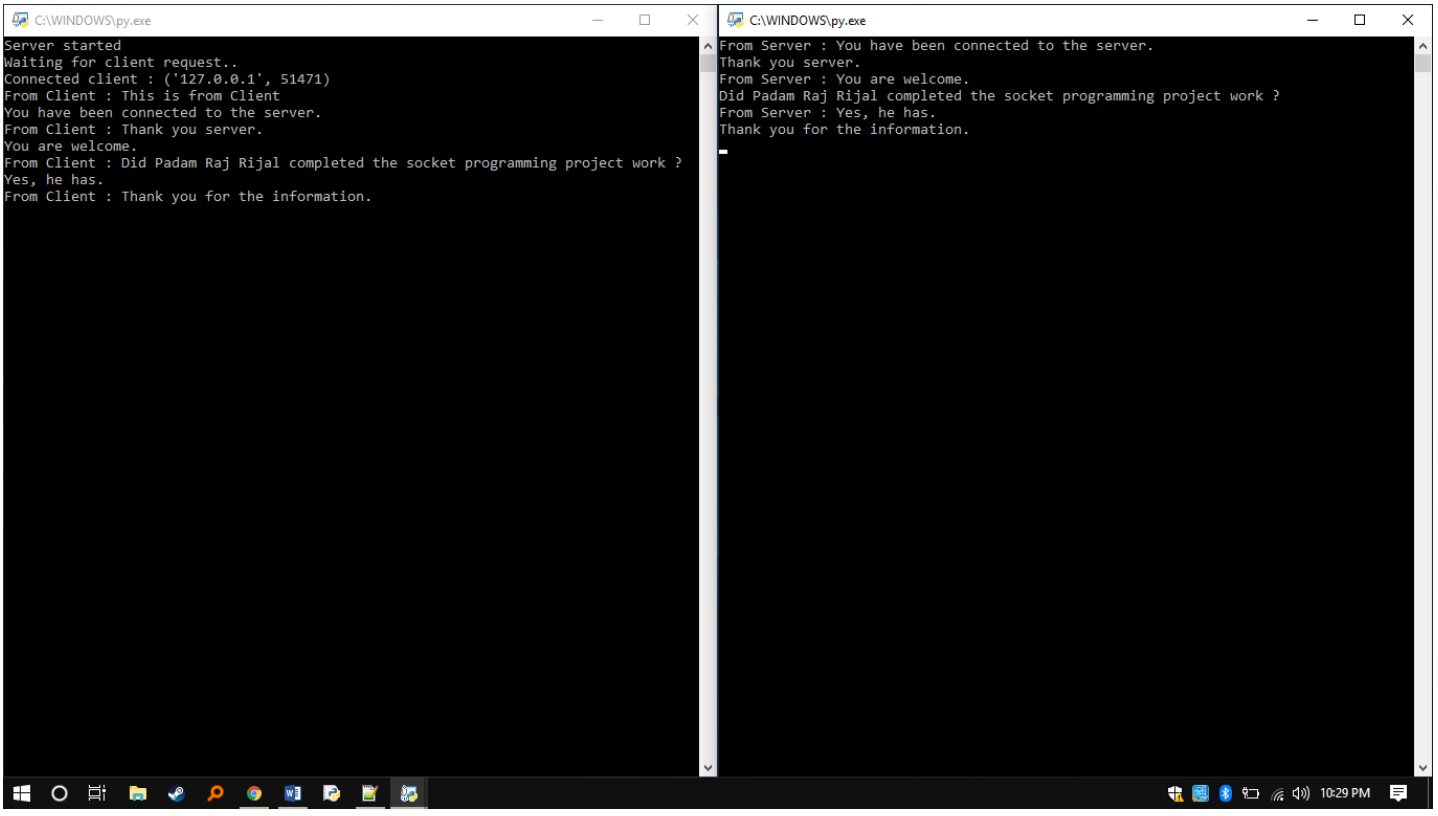

Fig4: communication between client and server

CHAPTER-8

CONCLUSION

In this project you can see that I have created two files client and server in python programming language. And we can see that we can communicate between the client and server. I wanted to do this project in python programming language because Mr. Yogesh sir told us that python has a lot of potential in doing any kind of project. It can be used in machine learning, neural net, voice recognition, network programming and web development.

So, I choose python language for the socket programming project work. And competed the project.

REFERENCES

1. http://net-informations.com/python/net/socket.htm
2. Computer Networking A Top-Down Approach by James F. Kurose and Keith W. Ross