Network Architecture Project Report

Team Members:

Pragathi Thammaneni

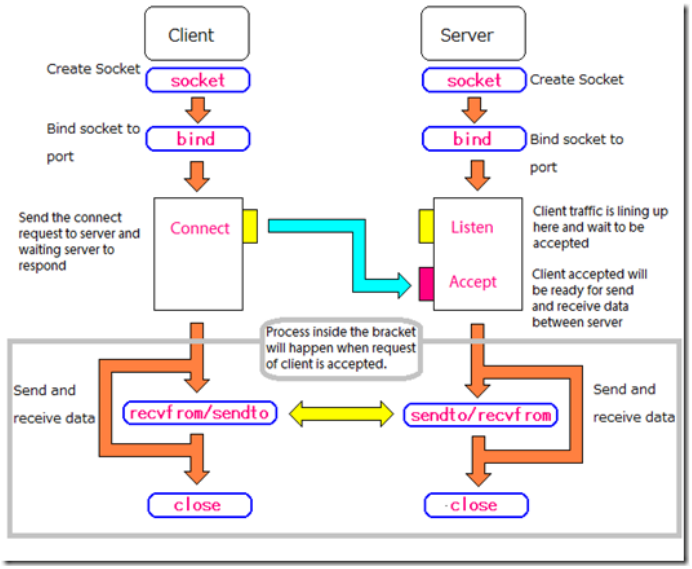
Sridevi Mallipudi

CLIENT & SERVER INTERACTION BY USING SOCKET PROGRAMMING

Introduction:

A network socket is an internal endpoint for sending or receiving data within a node on a computer network. Concretely, it is a representation of this endpoint in networking software, such as an entry in a table, and is a form of system resource.

Client & Server workflow:

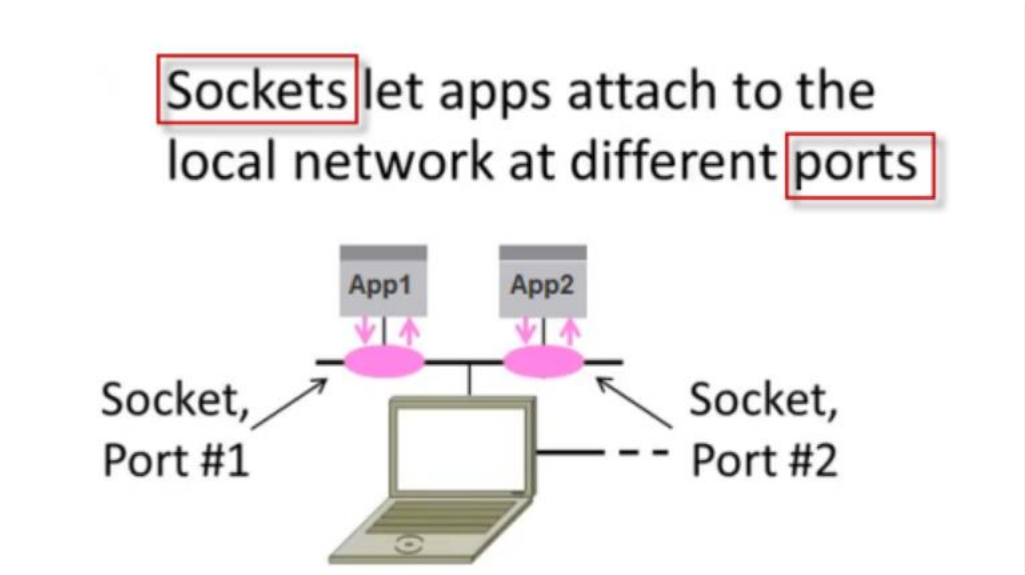


Motivation:

The client-server model solves the rendezvous problem by asserting that in any pair of communicating applications, one side must start execution and wait (indefinitely) for the other side to contact it. The solution is important because TCP/IP does not respond to incoming communication requests on its own.

Because TCP/IP does not provide any mechanisms that automatically create running programs when a message arrives, a program must be waiting to accept communication before any requests arrive.

Thus, to ensure that computers are ready to communicate, most system administrators arrange to have communication programs start automatically whenever the operating system boots. Each program runs forever, waiting for the next request to arrive for the service it offers. Thus, this project gives the client-server paradigm divides communicating applications into two broad categories, depending on whether the application waits for communication or initiates it. This project provides a concise, comprehensive definition.



Contribution:

We collectively implemented the Socket Programming in python for Client Server Interaction.

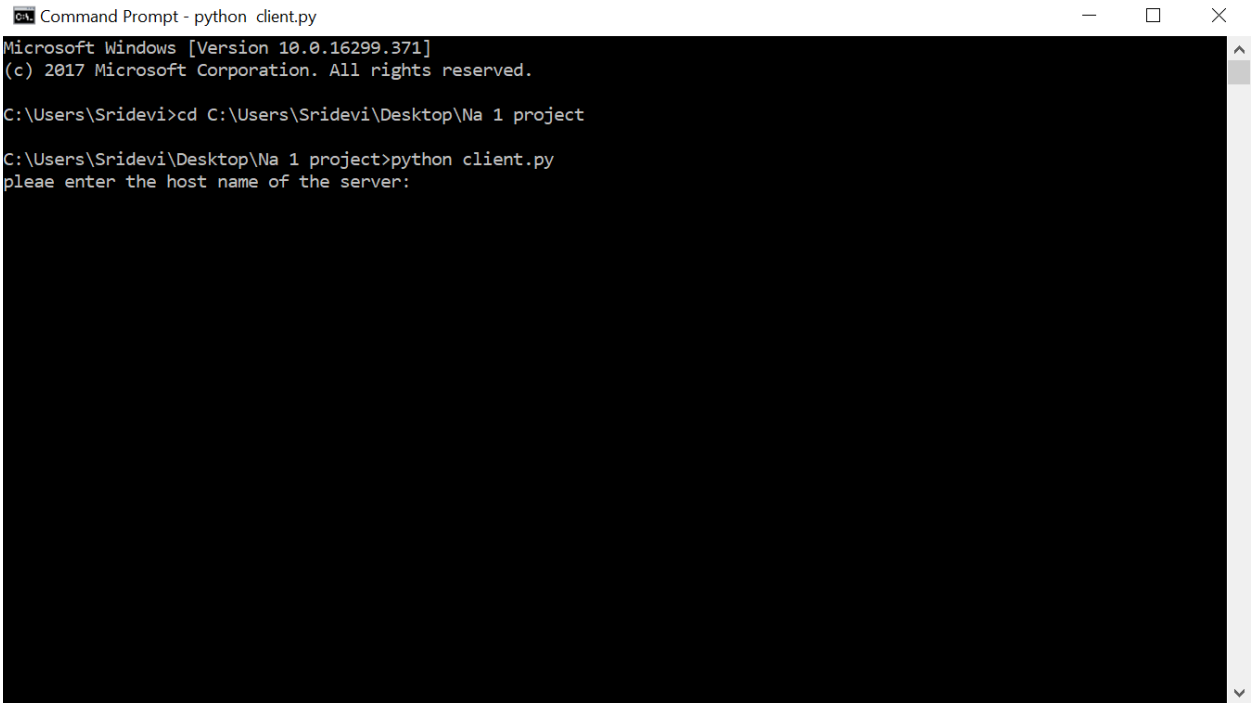
Tools used:

Python 3.7

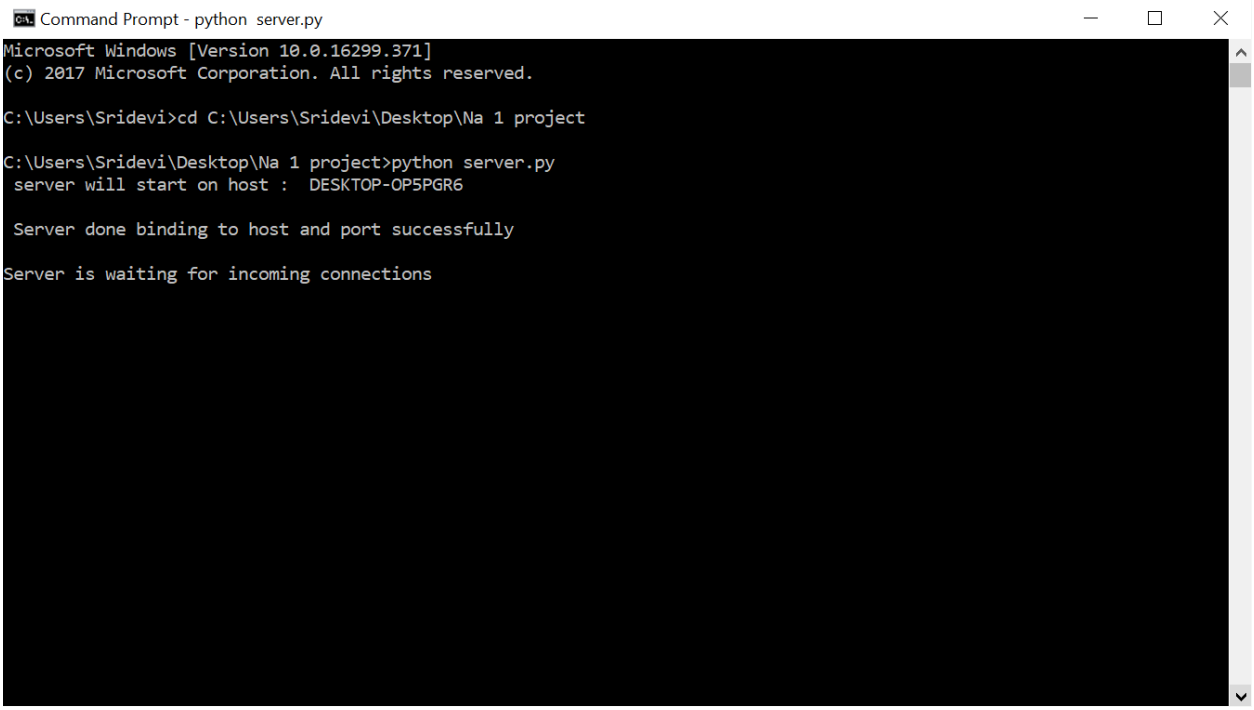
Outcome:

The client-server paradigm divides communicating applications into two broad categories, depending on whether the application waits for communication or initiates it.

Client:

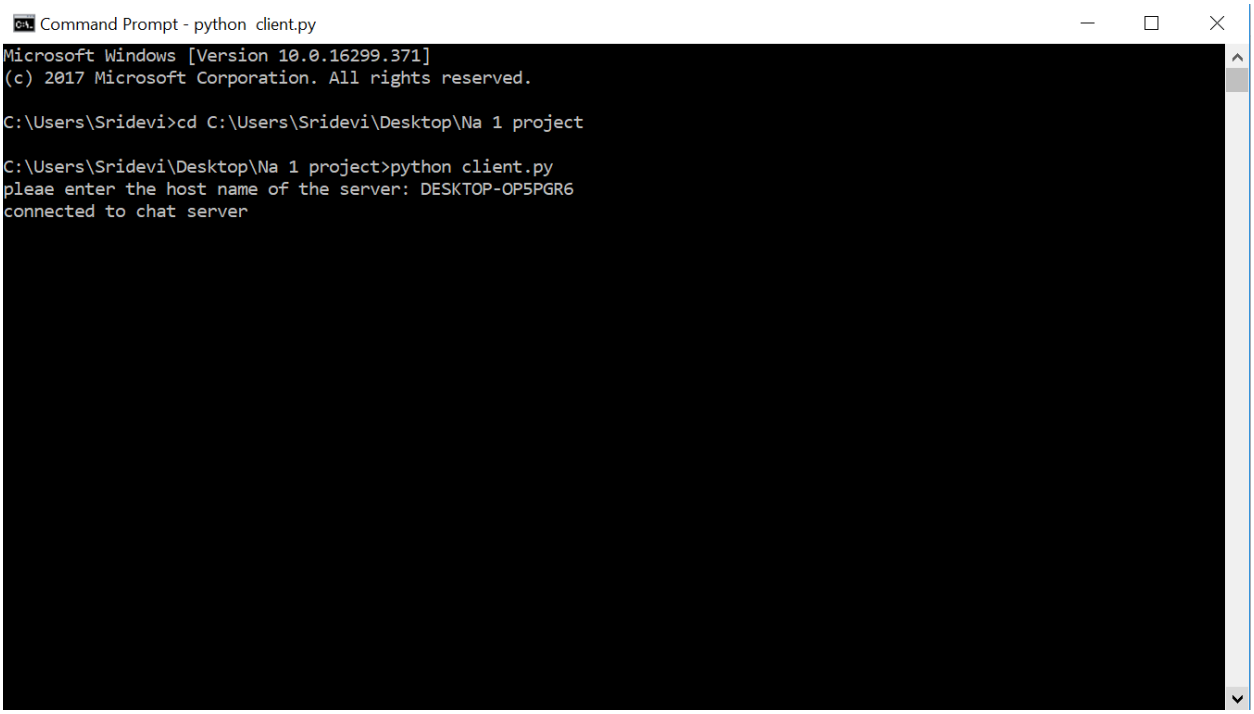


Server:

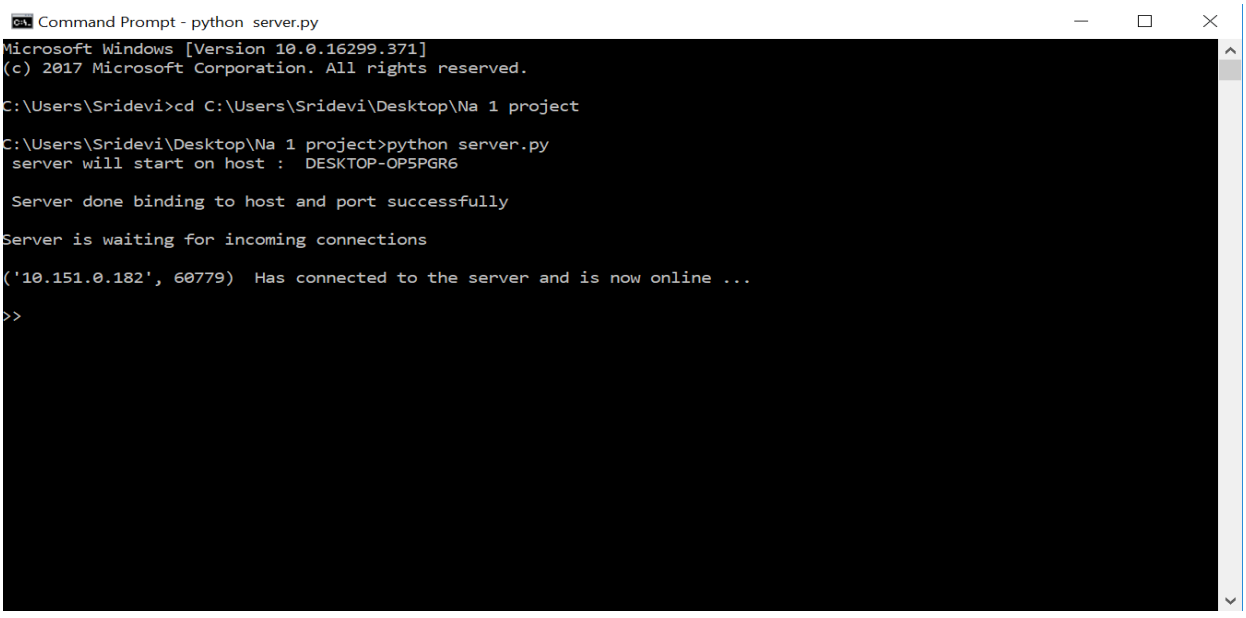


Client server interaction by sending messages -Sockets:

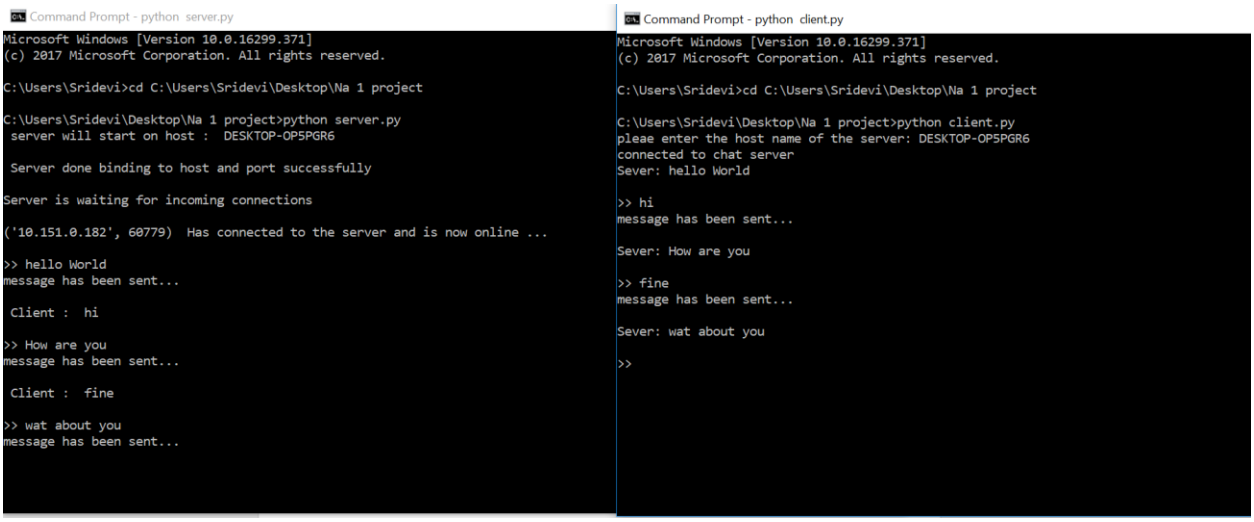
Please enter the host name of the server: DESKTOP-OP5PGR6 thus a message will be displayed about the status of the connection establishment.



Status at server side:



Message transfer:



Conclusion:

The initial aim of this project was to develop a small easily configurable and standalone client & server which interacts while the connection establishment**.** Thus, this project helps to get hands on experience with socket programming in real time.