Lab 3 Network Programming

Monday, March 4, 2019 12:18 PM

- We'll write a program on the client, and a program on the server, they're communicating with each other.
- There're two types of transmission protocols between client and server:-
  - TCP ===> connection-oriented protocol
  - UDP ===> connectionless protocol
- We're going to use a connection-oriented protocol.

What's the socket???

IP, and port numbers.

Isn't IP enough for identifying the source and destination???

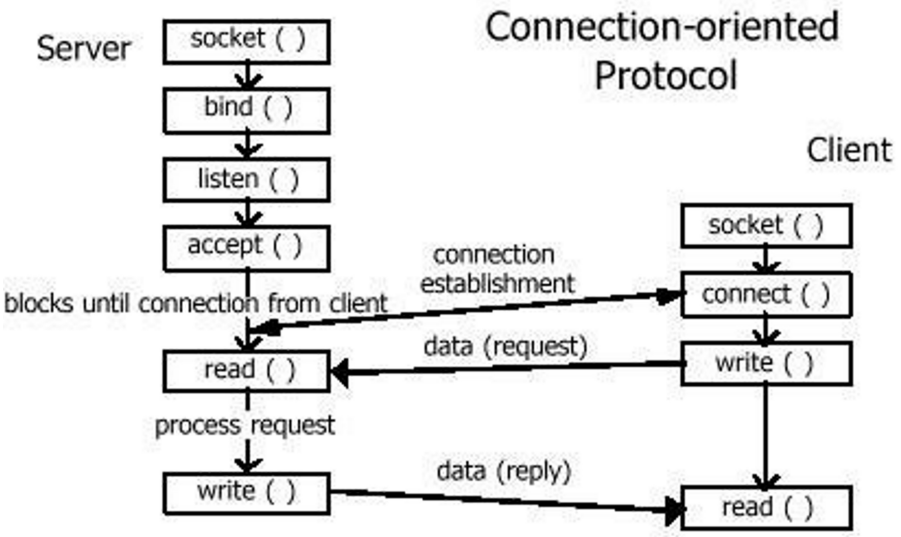
No, port number helps multiple services "applications".

What's the difference between windows, and Linux in this manner???

all port numbers in windows are open by default.

closed in Linux

What's operation of providing a service by a server for a client???



In the server:-

1. Invokes the socket method, which creates an unbounded socket ===> just generation of a socket number, that's not bounded with a specific IP and port numbers.
2. Invokes the bind method, It bounds the IP with the port number.
   - IP is a general identifier of the server, but the port number is an identifier of the application or service of the server.
3. Invoke the listen method for waiting for a request from the client.

In the client:-

1. Invokes the socket method for the same purpose.
2. Invokes the connect function for sending a request session.

In the server:-

4. Invokes the accept method, It accepts the request comes from the client.

In the client:-

3. Then R/W functions are created.
4. Then terminating "close".

What's the difference between socket number and session number???

- Socket is used for binding, listening, and accepting connection.
- Session is used for sending, and receiving between client and server.
    - It's used to distinguish between multiple clients being served by the server.

- When a client finishes taking his service, The client closes the socket.
- When a server finishes giving his service to a certain client, it closes the session, but leaving the socket to stay listening for other requests.

Programming

FOR SERVER:-

from Socket import *

s "file descriptor" **= socket (AF_INET, SOCK_STREAM)**

socket ("ip family" ===> IPv4 or 6, "protocol"===> TCP/UDP "conection-oriented/Connectionless")

host = "127.0.0.1" ===> identifying the IP of the server that the client could request for a service by.

port = 7000 ===> of the server

s.bind ((host, port))

s.listen(5"Any number" ===> Queue list**)** ===> that affects the resources.

while True:

c, ad=s.accept() ===> acceptance on socket.

What's c ???

Number of established session between the server, and a certain client, which's used to distinguish between sessions with different clients.

What's ad ???

an abbreviation for address

it contains the socket info about the other-end.

print ("connection from", ad[0])

x=c.recv(500"maximum size could be received"**)** ===> receiving the message sent by the client in a variable x.

c is the variable that the communication is done through.

print (x) ===> printing the received message from the client.

FOR CLIENT:-

from socket import *

s=socket(AF_INET,SOCK_STREAM)

host = "127.0.0.1" ===> of the server

port=7000 ===> of the server

s.connect((host, port))

s.send (b'hello') ===> b is an encoding mechanism ==> like uni-code-8, we're going to study in details,

hello is a message sent by the client for the server, the server will receive it in the variable x and print.

s.close()

Execution

1. Open the CMD
2. Choose the path of the server.py file
3. Write **python server.py**
4. Open another CMD
5. Choose the path of the client.py file
6. Write **python client.py**
7. Check the server's CMD.