Experiment 8

Aim: - To create a file transfer program using FTP protocol.

Steps for server: -

1. We import socket module which helps in establishing the client/server communication. Import Subprocess.
2. Bind the host from the port address using the bind() function.
3. We receive the data from the client using recv() and checking what command is required to be executed.
4. Using subprocess.run() to implement the remote commands given by the client.

Steps for Client: -

1. We import socket module which helps in establishing the client/server communication.
2. Connect the client and server and take the input of the command you want to send to the server.
3. Receive the data if exit is the command close the connection.

Code for Server:-

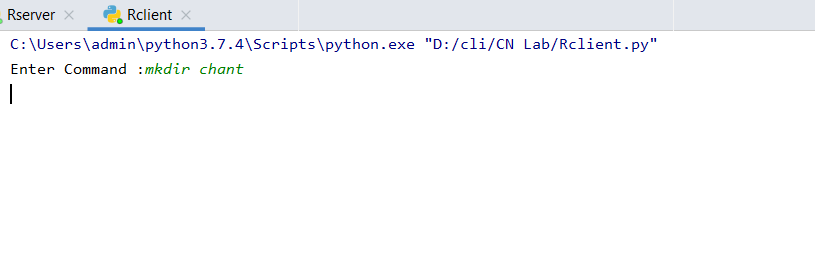
**import** socket  
**import** sys  
**import** subprocess  
s=socket.socket()  
host=socket.gethostname()  
port=1234  
s.bind((host,port))  
s.listen(5)  
print(**"Waiting..."**)  
conn,addr=s.accept()  
print(**"Recieving From"**,addr)  
**while True**:  
 data=conn.recv(1024).decode()  
 print(**"Client:"**,data)  
 **if** data==**"Exit" or not** data:  
 print(**"Server Exiting"**)  
 **break  
 else**:  
 p=subprocess.run(data,capture\_output=**True**,shell=**True**)  
 response=p.stdout  
 conn.send(response)  
s.close()

Code for Client:-

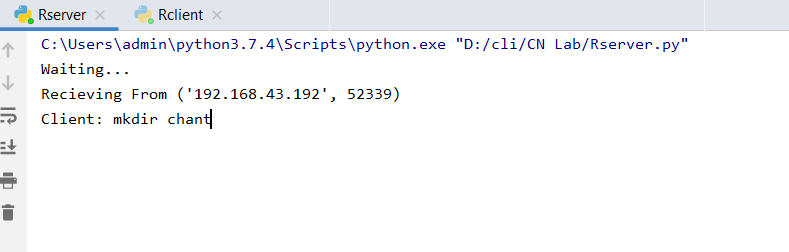
**import** socket  
**import** sys  
**import** subprocess  
s=socket.socket()  
host=socket.gethostname()  
port=1234  
s.connect((host,port))  
message=input(**"Enter Command :"**)  
s.send(message.encode())  
**while True**:  
 data=s.recv(1024).decode()  
 print(**"server"**,data)  
 **if** data==**"exit" or not** data:  
 print(**"Client exiting"**)  
 **break** msg=input(**"enter client message:"**)  
 s.send(msg.encode())  
s.close()

Console Screenshots : -

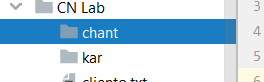
Client Console:-



Server Console:-



Created Directory : -



Result : -Remote command execution was successfully done using subprocess.