```
   • X11-forwarding  :  ✓  (remote display is forwarded through SSH)

 ➤ For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Aug 31 01:46:13 2024 from 10.0.2.2
mininet@mininet-vm:~$ dir
CST311  mininet  oflops  oftest  openflow  pox
mininet@mininet-vm:~$ cd CST311
mininet@mininet-vm:~/CST311$ dir
Basic_Socket_Programming
mininet@mininet-vm:~/CST311$ cd Basic_Socket_Programming/
mininet@mininet-vm:~/CST311/Basic_Socket_Programming$ dir
Individual_Programming_Assignment\ _1_Basic_Socket_Programming.pdf  RahimSiddiq_Programming_Assignment
\ _1_Basic_Socket_Programming.pdf  udp_client.py  udp_server.py
mininet@mininet-vm:~/CST311/Basic_Socket_Programming$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=865>
<Host h2: h2-eth0:10.0.0.2 pid=869>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=874>
<Controller c0: 127.0.0.1:6653 pid=858>
mininet> dir
*** Unknown command: dir
mininet> sh dir
Individual_Programming_Assignment\ _1_Basic_Socket_Programming.pdf  RahimSiddiq_Programming_Assignment
\ _1_Basic_Socket_Programming.pdf  udp_client.py  udp_server.py
mininet> h1 dir
Individual_Programming_Assignment\ _1_Basic_Socket_Programming.pdf
RahimSiddiq_Programming_Assignment\ _1_Basic_Socket_Programming.pdf
udp_client.py
udp_server.py
mininet> h2 dir
```

```
Individual_Programming_Assignment\ _1_Basic_Socket_Programming.pdf
RahimSiddiq_Programming_Assignment\ _1_Basic_Socket_Programming.pdf
udp_client.py
udp_server.py
mininet>  h1 python3 udp_server.py &
mininet> h2 python3 udp_client.py
Input sentence: csumb
Received: CSUMB
Input sentence: cst
Received: CST
Input sentence: 311
Received: 311
Input sentence: is pRettY FuN !!!
Received: IS PRETTY FUN !!!
Input sentence: doNE
mininet> cat udp_client.py
*** Unknown command: cat udp_client.py
mininet> sh cat udp_client.py
"""
File Name: udp_client.py
Author: Rahim Siddiq
Date: 08/30/2024
CST311 Programming Assignment 1
Description: UDP client script sends a message to a server and receives a response
"""

import socket

HOST = "10.0.0.1"  # todo: specify the server's hostname or IP address inside the quotes
PORT = 12345 # todo: specify the port number used by the server

with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
    s.connect((HOST, PORT))

    while True:
        message = input("Input sentence: ")
        if message.lower() == 'done':
            break

        byte_msg = message.encode('utf-8')
        s.sendall(byte_msg)
        data = s.recv(1024)

        print("Received: {}".format(data.decode('utf-8')))mininet>
mininet> sh cat udp_server.py
"""
File Name: udp_server.py
Author: Rahim Siddiq
Date: 08/30/2024
CST311 Programming Assignment 1
Description: UDP server script - listens on a specified ip and port. The server receives a message
    from a client, converts it to uppercase and returned the modified message back to the client.
"""

import socket

HOST = "10.0.0.1"  # todo: specify the correct hostname of IP address to communicate with the server.
PORT = 12345  # todo: specify the correct port number to communicate with the server.

# open a UDP socket
with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
    s.bind((HOST, PORT))
    print('Server listening on {}:{}'.format(HOST, PORT))
    while True:
        data, addr = s.recvfrom(1024)
        if not data:
            break
        s.sendto(data.upper(), addr)mininet>
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 177.239 seconds
mininet@mininet-vm:~/CST311/Basic_Socket_Programming$
```