PROJECT 1

To understand more about sockets, I first started off creating a single threaded server and client.

Server:

1. I created a socket object. **socket.AF_INET (IPv4)** is used for the address family and **socket.SOCK_STREAM** is the socket type. It means that the socket is of the type TCP.
2. I bound it to a host & port and then made it listen for a client. Host is local host (127.0.0.1)
3. Wrote the code where the server **1**-accepts the data from the client.

   2-processes it (I have used **extended slicing** to reverse the string).

   3-sends the data back to the client.

Client:

1. Since socket is the end point of communication between the client and server, I first created the socket object.
2. I connected the socket to the host and port of that on the server.
3. Then, I sent the data from client to server.
4. Then received data from the server which the client displays.

Multi-threading:

But since the server above is a single threaded server, when I try to connect multiple clients, it won't work. So I added **multi-threading** functionality to the server. I did this using the function **start_new_thread( ).** Now when I connect multiple clients, the server creates a thread for every new client request.

I got to know that recv( ) function accepts bytes object data. I used encode( ) and decode( ) functions to convert the Unicode string data to bytes object data and vice versa respectively.

I achieved the above functionality by importing the socket and thread libraries and using the following API functions:

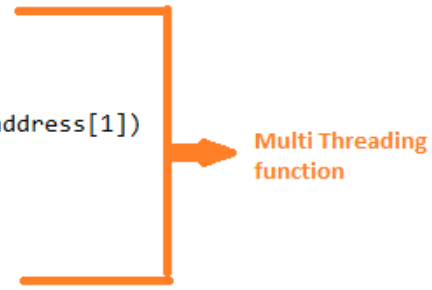
- socket( )
- bind( )
- listen( )
- accept( )
- send( )
- recv( )

To disconnect a client from the server, type "**exit**" on the client window.

Screenshots:

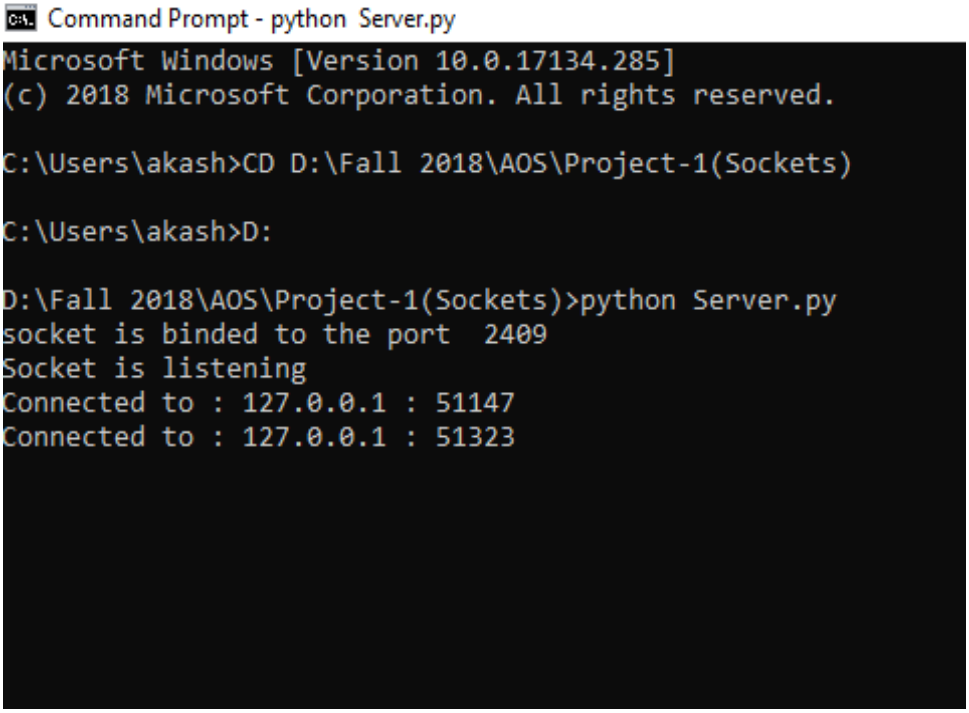
This is the server file which has the multithreading capability

```
import socket
from _thread import *
import threading
def threadFunc(conn,address):
    while True:
        data = conn.recv(1024)
        if not data:
            print('Disconnected from :', address[0], ':', address[1])
            break
        data = data[::-1]
        conn.send(data)
    conn.close()


host = "127.0.0.1"
port = 5000
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((host, port))
print("socket is binded to the port ", port)
s.listen(5)
print("Socket is listening")

while True:

    conn, address = s.accept()
    print('Connected to :', address[0], ':', address[1])
    start_new_thread(threadFunc, (conn,address))
s.close()
```



Connecting to client and server:

Server: This is the server; we can see that there are multiple clients connected to the server



Client1: I connected a client to the server. The connection was successful

Command Prompt - python Client.py

```
Microsoft Windows [Version 10.0.17134.285]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\akash>cd D:\Fall 2018\AOS\Project-1(Sockets)

C:\Users\akash>d:

D:\Fall 2018\AOS\Project-1(Sockets)>python Client.py
Enter a String: ADVANCE
Reversed string from server:ECNAVDA
Enter a String:
```

Client 2: I connected another client to the server. Since the server is multithreaded, it handled the second client request too.

Command Prompt - python Client2.py

```
Microsoft Windows [Version 10.0.17134.285]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\akash>cd D:\Fall 2018\AOS\Project-1(Sockets)

C:\Users\akash>d:

D:\Fall 2018\AOS\Project-1(Sockets)>python Client2.py
Enter a String: AKASH
Reversed string from server:HSAKA
Enter a String:
```

To disconnect from the server, type exit on the client window