

## Assignment 3

---

**Name:** Diptangshu Dey

**Roll No:** 20CS8018

---

1. Write a client and server program using socket programming in python to create a calculator of centralized and distributed manner.

Client Code:

```
import socket

host = 'localhost'
port = 9999
client = socket.socket()
client.connect((host, port))

print("Select an operation")
print("1. Add")
print("2. Subtract")
print("3. Multiply")
print("4. Divide")
print("5. Power")

sel = input("Enter: ")
x = input("Enter X: ")
y = input("Enter Y: ")
string = sel + "#" + x + "#" + y

client.send(string.encode())
data = client.recv(1024).decode()
print("Result = {}".format(data))
client.close()
```

Centralised Server Code:

```
import socket

host = 'localhost'
port = 9999
server = socket.socket()
server.bind((host, port))
server.listen(1)
print("Server Started")
print("Waiting for connection...")
```

```

while True:
    c, addr = server.accept()
    data = c.recv(1024).decode()
    t = data.split("#", 3)
    opr = int(t[0])
    num1 = int(t[1])
    num2 = int(t[2])

    res = 0
    if opr == 1:
        res = num1 + num2
        print("{} + {} = {}".format(num1, num2, res))
    elif opr == 2:
        res = num1 - num2
        print("{} - {} = {}".format(num1, num2, res))
    elif opr == 3:
        res = num1 * num2
        print("{} * {} = {}".format(num1, num2, res))
    elif opr == 4:
        res = num1 / num2
        print("{} / {} = {}".format(num1, num2, res))
    elif opr == 5:
        res = num1 ** num2
        print("{} ^ {} = {}".format(num1, num2, res))
    else:
        res = 0

    c.send(str(res).encode())
    c.close()

```

### Centralised Server Output:

The image shows two terminal windows side-by-side. The left window shows the server output for the 'cen\_calc\_ser.py' script, which displays 'Server Started', 'Waiting for connection...', and the results of calculations: '10 + 5 = 50' and '20 / 2 = 10.0'. The right window shows the client output for the 'calc\_cli.py' script, which displays a menu of operations (Add, Subtract, Multiply, Divide, Power), user input for operation 3, values X=10 and Y=5, and the result 'Result = 50'. It then shows another menu, user input for operation 4, values X=20 and Y=2, and the result 'Result = 10.0'.

### Desentralised Main Server Code:

```

import socket

s = socket.socket()
s.bind(('localhost', 9999))
s.listen(3)
print("Waiting for connection...")

while True:
    c, addr = s.accept()

```

```

data = c.recv(1024).decode()
p = data.split("#", 3)
opr = int(p[0])
res = None

s1 = socket.socket()
s1.connect(('localhost', 9999-opr))
s1.send(data.encode())

while res == None:
    res = s1.recv(1024).decode()

print(res)
print("result: {}".format(res))

c.send(str(res).encode())
c.close()

```

#### Decentralised Add Server Code:

```

import socket

s = socket.socket()
s.bind(('localhost', 9998))
s.listen(3)

while True:
    c, addr = s.accept()
    data = c.recv(1024).decode()
    p = data.split('#')
    x = int(p[1])
    y = int(p[2])
    res = str(x + y)
    print(res)
    c.send(res.encode())
    c.close()

```

#### Decentralised Subtract Server Code:

```

import socket

s = socket.socket()
s.bind(('localhost', 9997))
s.listen(3)

while True:
    c, addr = s.accept()
    data = c.recv(1024).decode()
    p = data.split('#')

```

```
x = int(p[1])
y = int(p[2])
res = str(x - y)
print(res)
c.send(res.encode())
c.close()
```

#### Decentralised Multiply Server Code:

```
import socket

s = socket.socket()
s.bind(('localhost', 9996))
s.listen(3)

while True:
    c, addr = s.accept()
    data = c.recv(1024).decode()
    p = data.split('#')
    x = int(p[1])
    y = int(p[2])
    res = str(x * y)
    print(res)
    c.send(res.encode())
    c.close()
```

#### Decentralised Division Server Code:

```
import socket

s = socket.socket()
s.bind(('localhost', 9995))
s.listen(3)

while True:
    c, addr = s.accept()
    data = c.recv(1024).decode()
    p = data.split('#')
    x = int(p[1])
    y = int(p[2])
    res = str(x / y)
    print(res)
    c.send(res.encode())
    c.close()
```

#### Decentralised Power Server Code:

```
import socket

s = socket.socket()
s.bind(('localhost', 9994))
s.listen(3)

while True:
    c, addr = s.accept()
    data = c.recv(1024).decode()
    p = data.split('#')
    x = int(p[1])
    y = int(p[2])
    res = str(x ** y)
    print(res)
    c.send(res.encode())
    c.close()
```

## Decentralised Output:

The image displays four terminal windows arranged in a 2x2 grid, illustrating a decentralized system. The left column shows the client's interaction, and the right column shows the servers' responses.

**Top Left Terminal (Client):** The user runs `python calc_cli.py`. The program prompts for an operation (1. Add, 2. Subtract, 3. Multiply, 4. Divide, 5. Power). The user enters '1', then '20' for X and '10' for Y. The program outputs 'Result = 30'.

**Top Right Terminal (Server):** The user runs `python dis_calc_dns_serv.py`. The program outputs 'Waiting for connection...', then '30', and finally 'result: 30'.

**Bottom Left Terminal (Client):** The user runs `python calc_cli.py` again. After the same prompts, the user enters '2' for the operation, '10' for X, and '5' for Y. The program outputs 'Result = 5'.

**Bottom Right Terminal (Servers):** This panel shows two separate server processes. The top one (DNS server) outputs 'Waiting for connection...', '30', and 'result: 30'. The bottom one (Sub server) outputs '5'.

```

A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> python calc_cli.py
Select an operation
1. Add
2. Subtract
3. Multiply
4. Divide
5. Power
Enter: 3
Enter X: 10
Enter Y: 3
Result = 30
A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> _

```

```

A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> python dis_calc_dns_serv.py
Waiting for connection...
30
result: 30
-

```

```

A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> python dis_calc_mul_ser.py
30
-

```

```

A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> python calc_cli.py
Select an operation
1. Add
2. Subtract
3. Multiply
4. Divide
5. Power
Enter: 3
Enter X: 10
Enter Y: 3
Result = 30
A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> python calc_cli.py
Select an operation
1. Add
2. Subtract
3. Multiply
4. Divide
5. Power
Enter: 4
Enter X: 20
Enter Y: 4
Result = 5.0
A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> _

```

```

A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> python dis_calc_dns_serv.py
Waiting for connection...
30
result: 30
5.0
result: 5.0
-

```

```

A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> python dis_calc_div_ser.py
5.0
-

```

```

A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> python calc_cli.py
Select an operation
1. Add
2. Subtract
3. Multiply
4. Divide
5. Power
Enter: 3
Enter X: 10
Enter Y: 3
Result = 30
A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> python calc_cli.py
Select an operation
1. Add
2. Subtract
3. Multiply
4. Divide
5. Power
Enter: 4
Enter X: 20
Enter Y: 4
Result = 5.0
A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> python calc_cli.py
Select an operation
1. Add
2. Subtract
3. Multiply
4. Divide
5. Power
Enter: 5
Enter X: 2
Enter Y: 10
Result = 1024
A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> _

```

```

A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> python dis_calc_dns_serv.py
Waiting for connection...
30
result: 30
5.0
result: 5.0
1024
result: 1024
-

```

```

A hobbist ~/MyFiles/Assignments/6th_Sem/CSS652_Networks_Lab/Lab_3 -> python dis_calc_pow_ser.py
1024
-

```