

EXP 1- SIMPLEX COMMUNICATION

Server side:

```
import socket
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.bind(('127.0.0.1',8000))
data,addr=s.recvfrom(1024)
print(data)
```

Client side

```
import socket
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.sendto(str.encode("hello"),('127.0.0.1',8000))
print('Message sent')
```

EXP 2- ECHOSERVER AND ECHOCLIENT

Server side

```
import socket
s= socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.bind(('127.0.0.1',8000))
s.listen(1)
print("waiting")
conn,addr=s.accept()
print("conneection established",addr)
data=conn.recv(1024)
print("received",data)
conn.sendall(data)
conn.close()
s.close()
```

Client side

```
import socket
c=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
c.connect(('127.0.0.1',8000))
msg=str(input('enter msg'))
print('sending',msg)
c.sendall(msg.encode())
data=c.recv(1024).decode()
print('echo',data)
c.close()
```

EXP 3- DATE & TIME

Server side

```
import socket
import time
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.bind(('127.0.0.1',8000))
s.listen(1)
conn,addr=s.accept()
print("got a connection from ", addr)
currenttime=time.ctime(time.time())+"\r\n"
conn.send(currenttime.encode())
#conn.close()
```

Client side

```
import socket
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(('127.0.0.1',8000))
tm=s.recv(1024).decode()

print("the time is ",tm)
#s.close()
```

EXP 4 -HALF DUPLEX

Server side

```
import socket
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.bind(('127.0.0.1',8000))
print('waiting')
while True:
    data,addr=s.recvfrom(1024)
    print('msg from client is',data.decode())
    data1=str(input("enter msg:"))
    s.sendto(data1.encode(),('127.0.0.1',8000))
    print('msg sent')
```

Client side

```
import socket
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
while True:
    msg=str(input('enter msg:'))
    s.sendto(str.encode(msg),('127.0.0.1',8000))
    data,addr=s.recvfrom(1024)
```

```
print('response is:',data.decode())
```

EXP 5- FULL DUPLEX

Server side

```
from concurrent.futures import thread
from string import whitespace
import socket

import time
import threading

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

server.bind(('localhost', 8000))

server.listen(1)

print("Server Up and Running")

conn, addr = server.accept()

print("Connection Accepted", conn)

def getmsg():
    while True:
        msg = conn.recv(1000).decode('utf-8')
        print("Client: ", msg)
        # print("\n")

        time.sleep(0.5)

def sendmsg():
    while True:
        msg = input("")
        # print("\n")

        conn.sendall(msg.encode())
        time.sleep(0.5)

thread1 = threading.Thread(target=getmsg)
thread1.start()

thread2 = threading.Thread(target=sendmsg)
thread2.start()
```

Client side

```
from concurrent.futures import thread
from pydoc import cli
from string import whitespace
import socket

import time
import threading

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

client.connect(('localhost', 8000))

def getmsg():
    while True:
        msg = client.recv(1000).decode('utf-8')
        print("Server: ", msg)
        time.sleep(0.5)

def sendmsg():
    while True:
        msg = input("")
        # print("\n")

        client.sendall(msg.encode())
        time.sleep(0.5)

thread2 = threading.Thread(target=sendmsg)
thread2.start()

thread1 = threading.Thread(target=getmsg)
thread1.start()
```

EXP 6-ARP PROTOCOL

Server side

```
mac_ip={
    "10.0.2.5": "00:16:3e:99:0b:db",
    "10.0.5.5": "00:16:4e:99:0b:db",
    "11.0.2.5": "00:16:5e:99:0b:db",
    "15.0.2.5": "00:15:7c:99:0b:db",
    "17.0.2.5": "00:20:3d:99:0b:db"
}

import requests
url='http://ptsv2.com/t/3894f-1665744100/post'
```

```
r=requests.post(url,data=mac_ip)
print(r)
```

Client side

```
import json
import requests
url='http://ptsv2.com/t/3894f-1665744100/d/latest/json'
r=requests.get(url)
data=r.json()['FormValues']
for x in data:
    data[x] = data[x][0]
ip=input("Enter ip address:")
if ip in data.keys():
    print(data[ip])
else:
    print("ip address not found")
    mac=input("Enter corresponding mac address:")
    urlpost='http://ptsv2.com/t/3894f-1665744100/post'
    data[ip]=mac
    r=requests.post(urlpost,data=data)
```

EXP 7-DNS

```
import socket
dnsDomain ={
    '35.230.7.9' : 'ysyouthgct.in',
    '23.100.43.208' : 'www.ssd.com'
}
while True:
    print("enter choice \n1.ip \n2.url")
    ch=int(input())
    if ch==1:
        url=input("enter url")
        ip= socket.gethostbyname(url)

        print(ip)
        #dnsDomain.update({ip:url})

    if ch==2:
        ip=input("enter the ip address")
        host=socket.gethostbyaddr(ip)
        if host[2][0] in dnsDomain.keys():
            print("Dns Domain :",dnsDomain[host[2][0]])
```

EXP 8 -CRC

Server side

```
import socket

def xor(a, b):
    result = []

    for i in range(1, len(b)):
        if a[i] == b[i]:
            result.append('0')
        else:
            result.append('1')

    return ''.join(result)

def mod2div(divident, divisor):

    pick = len(divisor)

    tmp = divident[0: pick]

    while pick < len(divident):

        if tmp[0] == '1':
            tmp = xor(divisor, tmp) + divident[pick]

        else:
            tmp = xor('0'*pick, tmp) + divident[pick]

        pick += 1

    if tmp[0] == '1':
        tmp = xor(divisor, tmp)
    else:
        tmp = xor('0'*pick, tmp)

    checkword = tmp
    return checkword

def decodeData(data, key):

    l_key = len(key)
```

```

        appended_data = data.decode() + '0'*(1_key-1)
        remainder = mod2div(appended_data, key)

    return remainder

s = socket.socket()
print("Socket successfully created")

port = 12345

s.bind(('', port))
print("socket binded to %s" % (port))

s.listen(5)
print("socket is listening")

while True:

    c, addr = s.accept()
    print('Got connection from', addr)

    data = c.recv(1024)

    print("Received encoded data in binary format :", data.decode())

    if not data:
        break
    dat = data.decode()
    key = input("Enter divisor :")

    ans = decodeData(data, key)
    print("Remainder after decoding is->" + ans)

    dat = list(dat)
    val = input("Do u want to introduce error(1) : ")
    print(val)
    if val == "1":
        err = int(input("Error index : "))
        if dat[len(dat)-err] == "1":
            dat[len(dat)-err] = "0"
        else:
            dat[len(dat)-err] = "1"

```

```

dat = "".join(dat)

#temp = "0" * (len(key) - 1)
if dat == data.decode():
    c.sendto(("THANK you Data ->" + dat +
            " Received No error FOUND").encode(), ('127.0.0.1', 12345))
else:
    c.sendto(("Error in data->" + dat).encode(), ('127.0.0.1', 12345))

c.close()

```

Client side

```

import socket

def xor(a, b):

    result = []

    for i in range(1, len(b)):
        if a[i] == b[i]:
            result.append('0')
        else:
            result.append('1')

    return ''.join(result)

def mod2div(divident, divisor):

    pick = len(divisor)

    tmp = divident[0 : pick]

    while pick < len(divident):

        if tmp[0] == '1':

            tmp = xor(divisor, tmp) + divident[pick]

        else:

            tmp = xor('0'*pick, tmp) + divident[pick]

        pick += 1

```



```

        if tmp[0] == '1':
            tmp = xor(divisor, tmp)
        else:
            tmp = xor('0'*pick, tmp)

        checkword = tmp
        return checkword

def encodeData(data, key):

    l_key = len(key)

    appended_data = data + '0'*(l_key-1)
    remainder = mod2div(appended_data, key)

    codeword = data + remainder
    return codeword

s = socket.socket()

port = 12345

s.connect(('127.0.0.1', port))

input_string = input("Enter data you want to send->")

data =(''.join(format(ord(x), 'b') for x in input_string))
print("Entered data in binary format :",data)

key = input("Enter divisor :")

ans = encodeData(data,key)
print("Encoded data to be sent to server in binary format :",ans)
s.sendto(ans.encode(),('127.0.0.1', 12345))

print("Received feedback from server :",s.recv(1024).decode())

s.close()

```

EXP 9-SUBNETTING

```
ip=input("enter ip")
sub_mask=input("enter ")
#print("ip and submask is"ip+sub_mask)
sub_mask=int(sub_mask)
arr=ip.split(".")
print(arr)
binary=''
for i in arr:
    binary+=format(int(i),'08b')
print(binary)
index=32-sub_mask
list1=list(binary)
list2=list(binary)
for i in range(1,index+1):
    list1[-i]='0'
    list2[-i]='1'
print(list1)
print(list2)
temp1=''
temp2=''
for i in list1:
    temp1+=i
for i in list2:
    temp2+=i
print(temp1)
print(temp2)
bin1=[]
bin2=[]
for i in range(0,len(temp1),8):
    bin1.append(temp1[i:i+8])
    bin2.append(temp2[i:i+8])
print(bin1)
print(bin2)
ans1=''
ans2=''
for i in range(0,4):
    bin1[i]=int(bin1[i],2)
    bin2[i]=int(bin2[i],2)
    ans1+=str(bin1[i])
    if(i!=3):
        ans1+='.'
    ans2+=str(bin2[i])
    if(i!=3):
        ans2+='.'
print(bin1)
print(bin2)
print(ans1)
```

```
print(ans2)
```

EXP 10-UPLOAD AND DOWNLOAD

Download

```
import urllib.request, urllib.error, urllib.parse

print(" Beginning file download ")

url = input("Enter valid url : ")

response = urllib.request.urlopen(url)
webContent = response.read().decode('UTF-8')

print(webContent[0:2000])
```

BY IT GIRLS.....