Contents

[FTP 3](#_Toc42337935)

[Đếm ảnh trong thư mục HEADER.images 3](#_Toc42337936)

[Cookie 3](#_Toc42337937)

[Giao thức 4](#_Toc42337938)

[Header 4](#_Toc42337939)

[Chia mạng con 4](#_Toc42337940)

[DNS 5](#_Toc42337941)

[Lấy IP trong windows 5](#_Toc42337942)

[SMTP 6](#_Toc42337943)

[Gzip 7](#_Toc42337944)

[Đếm số ảnh trong 1 trang: 7](#_Toc42337945)

[TCP C# chung 7](#_Toc42337946)

[TCP python chung 7](#_Toc42337947)

[UDP C# chung 8](#_Toc42337948)

[UDP python chung 8](#_Toc42337949)

[TCPListener C# chung 9](#_Toc42337950)

[TCPClient C# chung 9](#_Toc42337951)

[Tổng hiệu 2 số (python tcp) 10](#_Toc42337952)

[Chuẩn hóa xâu 10](#_Toc42337953)

[TCP 10](#_Toc42337954)

[UDP 11](#_Toc42337955)

[Xdựng ctrình, client yêu cầu user nhập ‘r’ để bđ knoi đến sver. Sau khi svẻ nhận tín hiệu thì trả về các option (1.tru/2.cong). C nhận các option từ sv in ra màn hình. Sau đó yêu cầu user nhập a,b,c. Truyền gtrị cho sv để tính kquả, và trả về tổng hoặc hiệu. 11](#_Toc42337956)

[TCP 11](#_Toc42337957)

[UDP 12](#_Toc42337958)

[C# TCP 13](#_Toc42337959)

[UDP C# 14](#_Toc42337960)

[Sver lưu dssv gồm: msv,tên sv. Client gửi lời chào sver-sver gửi lại. Sver gửi thông điệp ‘nhập 1 để bđ’ if client gửi 1 thì server yêu cầu nhập msv. Sver nhận msv và trả về thông tin tương ứng với msv 15](#_Toc42337961)

[TCP 15](#_Toc42337962)

[UDP 16](#_Toc42337963)

[Client gửi UP(LO) và 1 xâu ký tự đến sver. Và nhận kq trả về. Sver hiện thông tin client, ktra lệnh nhận đc là gì? Nhận xâu. Thực hiện biến đổi xâu thành chữ hoa (UP) or thường(LO). Trả kq về client. 16](#_Toc42337964)

[TCP 16](#_Toc42337965)

[UDP 17](#_Toc42337966)

[Chia UDP 17](#_Toc42337967)

[Lý thuyết: 18](#_Toc42337968)

# FTP

from ftplib import FTP  
if \_\_name\_\_ == '\_\_main\_\_':  
 ftp = FTP('ftp.ibiblio.org')  
 print(ftp.getwelcome()) *# getwelcome(): trả về thông báo khi connect*  
 ftp.login()

# *nếu yêu cầu username, password: ftp=FTP('ftp.ibiblio.org','username','password')*  
 print("Thu muc hien thoi: ", ftp.pwd()) *# pwd(): trả về thư mục hiện thời*  
 *# ftp.cwd('/incoming/celestia/') # đưa đường dẫn thư mục: cwd(path)* dirftp = ftp.nlst() *#'incoming' # hien thị các file và thư mục: nlst()* f = open('file', "w")  
 read = ftp.retrlines('RETR '+'README',f.write)  
 ftp.quit()  
 print("tong so: ",len(dirftp)," file va thu muc")  
 for i in sorted(dirftp):  
 print(i) *# hien thi cac file va thu muc ben trong thu muc incoming  
 # download file: ASCII/Binary: ftp.retrlines: đọc file trên server  
 # ftp.retrlines("RETR "+filename, f.write)  
 # upload file: ftp.storbinary("STOR "+ filename,f)*

Đếm ảnh trong thư mục HEADER.imagesimgCount = 0  
extensions = ["jpg", "gif", "jpeg", "bmp", "png"]  
dirftp = ftp.nlst('/HEADER.images')  
for i in sorted(dirftp):  
 arr = i.split(".")  
 if (arr[len(arr) - 1]) in extensions:  
 imgCount += 1  
print(imgCount)  
ftp.quit()

# Cookie

from http.cookiejar import CookieJar  
from urllib.request import build\_opener, HTTPCookieProcessor  
import datetime  
cookie\_jar=CookieJar()  
op=build\_opener(HTTPCookieProcessor(cookie\_jar))  
# sử dụng opener để tạo http request  
op.open('http://www.github.com')  
cookies=list(cookie\_jar) # tạo ds để lấy từng cookie  
print(cookies)  
print(cookies[0].name)  
print(cookies[0].value)  
print(cookies[0].expires)  
print(datetime.datetime.fromtimestamp(cookies[0].expires))

# Giao thức

from urllib.request import urlopen  
from urllib.request import Request  
resp=urlopen('http://www.facebook.com') *# gửi yêu cầu mở facebook gán vào biến resp*print(resp.read()) *# đọc body web page bằng read() (đọc hết 1 lượt)*print(resp.status) *# Thong bao VD: 404: ko tim thay url*print(resp.url) *# Hien thi link*print(resp.readline()) *# doc 1 dong*

# Header

from urllib.request import Request  
from urllib.request import urlopen  
import gzip  
r=Request('http://www.python.org')  
r.add\_header('Accept-Language','vi') *# thêm header cho request*

*# r.add\_header('Accept-Encoding', 'gzip') nén/mã hóa*r1= urlopen(r) *#gửi*

print(r.get\_header('User-agent')) *# Lấy ra trình duyệt của user khi gửi request*

print(r1.readline()) *# doc 1 dong html*print(r1.read()) *# doc het tat ca html*print(r1.url) *# lấy ra url: facebook.com*print(r1.status) *# lấy ra status của website*print(r1.getheaders()) *# lấy ra header của website*print(r1.getheader('Content-Encoding')) *# ktra header*print(r1.getheader('Content-Type'))  *# print(r.header\_items())*  
f, p=r1.getheader('Content-Type').split(';') *#ktra header*print(p)

# Chia mạng con

*# 192.168.0.0, subnet mask: 255.255.255.0  
# 192.168.0.0/24, 192.168.0.0/26 -> 4 subnets, moi subnet co 64  
# subnet mask: 255.255.255.192, xác định địa chỉ đầu, địa chỉ cuối, broadcast*import ipaddress  
if \_\_name\_\_ == '\_\_main\_\_':  
 ip\_addr = '192.168.0.0'  
 c = int(input("Nhap so bit (24-30):"))  
 net = ip\_addr + '/' + str(c)  
 print("network address %s" % net)  
 network = ipaddress.ip\_network(net)  
 *# subnet mask mới* print("subnet mask: %s" % str(network.netmask))  
 dcdau = list(network.hosts())[0]  
 dccuoi = list(network.hosts())[-1] *# cuối cùng* print("dia chi dau %s, dia chi cuoi %s" % (dcdau, dccuoi))  
 *# broadcast* print("broadcast %s" % str(network.broadcast\_address))

# DNS

\*) C#: Using System.Net

static void Main(string[] args){

string hostName = Dns.GetHostName();

Console.WriteLine(“Local hostname: {0},hostName);

IPHostEntry myself = Dns.GetHostEntry(hostName);

foreach(IPAddress address in myself.AddressList){

Console.WriteLine(“IPAddress: {0}”,address.ToString()); }

Console.ReadLine(); }

\*)Python *# DNS (Domain Name System): ánh xạ các tên miền với địa chỉ IP*import netifaces *# pip install netifaces*  
iface = netifaces.interfaces() *# tìm các giao tiếp mạng*  
for ifc in iface: ipaddr = netifaces.ifaddresses(ifc)  
 if netifaces.AF\_INET in ipaddr: ipaddr\_d = ipaddr[netifaces.AF\_INET]  
 ipaddr\_d = ipaddr\_d[0]  
 print("Network interface {0}".format(ifc))  
 print("IP address {0}".format(ipaddr\_d['addr']))  
 print("netmask {0}".format(ipaddr\_d['netmask']))  
 gateway = netifaces.gateways() *# liệt kê tất cả các gateways và địa chỉ IP*  
 print("Default gateway {0}".format(gateway['default'][netifaces.AF\_INET][0]))

# Lấy IP trong windows

\*) C# (Console app (net. Framework))

* Thêm tham chiếu (add reference) System.Management vào project

static void Main(string[] args){

ManagementObjectSearcher query = new ManagementObjectSearcher(“select \* from Win32\_NetworkAdapterConfiguration where IPEnabled=’true’”);

ManagementObjectCollection queryCollection = query.Get();

foreach( ManagementObject mo in queryCollection){

string[] addresses = (string[])mo[“IPAddress”];

string[] subnets = (string[])mo[“IPSubnet”];

string[] defaultgateways = (string[])mo[“DefaultIPGateway”];

Console.WriteLine(“Network card: {0}”, mo[“Description”]);

Console.WriteLine(“MAC Address: {0}”, mo[“MACAddress”]);

foreach(string ipaddress in addresses){

Console.WriteLine(“IPAddress: {0}”, ipaddress); }

foreach(string subnet in subnets){

Console.WriteLine(“IPAddress: {0}”, ipaddress); }

if(defaultgetways != null)

foreach(string defaultgateway in defaultgateways){

Console.WriteLine(“Gateway: {0}”, defaultgateway); }

}

Console.ReadLine(); }

\*)Python 1 :

import socket  
def getip():  
 hostname = socket.gethostname() *# ktra tên máy*  
 hostip = socket.gethostbyname(hostname) *# ktra địa chỉ IP*  
 print("hostname",hostname)  
 print("IPAddress",hostip)  
getip()

\*)Python 2:

import ipaddress as ip  
if \_\_name\_\_ == '\_\_main\_\_':  
 net4 = ip.ip\_network('10.0.1.0/24') *# định nghĩa mạng net4:*  
 print(net4)  
 *# tìm netmask, địa chỉ mạng, địa chỉ quảng bá từ net4* print(net4.netmask)  
 print(net4.network\_address)  
 print(net4.broadcast\_address)  
 print(net4.num\_addresses)  
 print(net4.hosts)  
 print(net4.subnets())  
 print(net4.supernet())

# SMTP

*#Tạo header*from email.mime.multipart import MIMEMultipart  
from email.mime.text import MIMEText

from email.mime.image import MIMEImage  
msg=MIMEMultipart()  
msg['To']= 'b@gmail.com' *#địa chỉ nhận thư*msg['From']='a@gmail.com' *#địa chỉ gửi thư*msg['Subject'] = 'Chu de cua thu'  
 *#Tạo body*part = MIMEText('text','plain')  
message = 'Thao gui ne'  
part.set\_payload(message)

img = MIMEImage('1.jpg')

msg.attach(img)  
msg.attach(part)  
 *#Gửi thư*import smtplib  
SMTP\_SERVER = 'aspmx.l.google.com'  
SMTP\_PORT = 25  
session = smtplib.SMTP(SMTP\_SERVER,SMTP\_PORT)  
session.ehlo()  
session.sendmail(' dc gửi thư ',' dc nhận thư ',msg.as\_string()) # đã ghi ở trên  
session.quit()

# Gzip

*import gzip  
content= gzip.decompress(r1.read()) giải nén dữ liệu bằng gzip  
content.splitlines()[:10] hiện baon dòng*

# Đếm số ảnh trong 1 trang:

from urllib.request import urlopen,Request  
r=Request('https://vnexpress.net/')  
r1=urlopen(r)  
string = str(r1.read())  
count = string.count('.jpg') //png  
print(count)

# TCP C# chung

|  |  |
| --- | --- |
| **Server:**  Socket sk = new Socket  (AddressFamily.InterNetwork,  SocketType.Stream, ProtocolType.Tcp);  IPEndPoint dc = new  IPEndPoint(IPAddress.Any, 9050);  sk.Bind(dc); sk.Listen(10);  Console.WriteLine("Waiting...");  Socket client = sk.Accept();  string s = "Hello client";  byte[] dl = new byte[1024];  dl = Encoding.ASCII.GetBytes(s);  client.Send(dl, dl.Length, SocketFlags. None);  int kt; kt = client.Receive(dl); Console.WriteLine(Encoding.ASCII.GetString(dl, 0, kt)); Console.ReadLine();  sk.Close(); client.Close(); | **Client:**  Using System.Net;  Using System.Net.Sockets;  Socket sk = new  Socket(AddressFamily.InterNetwork,  SocketType.Stream, ProtocolType.Tcp);  IPEndPoint dcsv = new  IPEndPoint(IPAddress.Parse("127.0.0.1"), 9050);  sk.Connect(dcsv);  Byte[] dl = new byte[1024];  int kt;  kt = sk.Receive(dl);  Console.WriteLine(Encoding.ASCII.GetString(dl, 0, kt));  sk.Send(Encoding.ASCII.GetBytes("helo 1"));  Console.ReadLine(); sk.Close(); |

# TCP python chung

|  |  |
| --- | --- |
| **server:** khai báo socket như bên  sk.bind((host,port)) sk.listen(5)  client, addr\_client = sk.accept()  client.send(s.encode('utf-8'))  data = client.recv(1024).decode() print(data)  while True: data= client.recv(1024)  if not data or data.decode('utf-8') == ‘’:  break  else : s=…..  client.send(s.encode('utf-8'))  client.close() sk.close() | **client:**  sk = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  sk.connect((host,port))  data = sk.recv(1024) print(data)  sk. send(s.encode('utf-8'))  while True:  s = input(“Nhap du lieu: ”)  sk.send(s.encode('utf-8'))  data=sk.recv(1024)  print(data.decode('utf-8'))  sk.close() |

# UDP C# chung

|  |  |
| --- | --- |
| **Server:**  using System.Net;  using System.Net.Sockets;  IPEndPoint ipep = new  IPEndPoint(IPAddress.Any, 80);  Socket sk = new  Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);  sk.Bind(ipep); Console.WriteLine("Waiting....");  IPEndPoint client = new IPEndPoint(IPAddress.Any, 0);  EndPoint remote = (EndPoint)(client);  int kt;  byte[] dl = new byte[1024];  kt = sk.ReceiveFrom(dl, ref remote);  Console.WriteLine(Encoding.ASCII.GetString(dl, 0, kt));  string s = "Hello client";  dl =  Encoding.ASCII.GetBytes(s);  sk.SendTo(dl, dl.Length, SocketFlags.None, remote);  while (true) {  dl = new byte[1024];  kt = sk.ReceiveFrom(dl, ref remote); Console.WriteLine(Encoding.ASCII.GetString(dl, 0, kt));  sk.SendTo(dl, kt, SocketFlags.None, remote); }  sk.Close(); | **Client:**  byte[] dl = new byte[1024];  IPEndPoint dc = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 80);  Socket sk = new  Socket(AddressFamily.InterNetwork,  SocketType.Dgram, ProtocolType.Udp);  string s = "Hello server";  dl = Encoding.ASCII.GetBytes(s);  sk.SendTo(dl, dl.Length, SocketFlags.None, dc);  IPEndPoint server = new  IPEndPoint(IPAddress.Any, 0);  EndPoint remote = (EndPoint)server;  dl = new byte[1024];  int kt = sk.ReceiveFrom(dl, ref remote);  Console.WriteLine(Encoding.ASCII.GetString(dl, 0, kt));  while (true) {  string input = Console.ReadLine();  if (input == "exit")  break;  sk.SendTo(Encoding.ASCII.GetBytes  (input), input.Length, SocketFlags.None, remote);  dl = new byte[1024];  kt = sk.ReceiveFrom(dl, ref remote);  Console.WriteLine(Encoding.ASCII.GetString(dl, 0, kt)); }  sk.Close(); |

# UDP python chung

|  |  |
| --- | --- |
| **server:**  sk = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  sk.bind((host,port))  data, addr = sk.recvfrom(1024) // nhan  print(data.decode('utf-8'))  sk.sendto(s.encode('utf-8'), addr) // gui  sk.close() | **client:**  sk = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  sk.sendto(s.encode('utf-8'), (host,port))  data, addr = sk.recvfrom(1024)  print(data.decode('utf-8'))  sk.close() |

# TCPListener C# chung

|  |  |
| --- | --- |
| **Server**  using System.Net;  using System.Net.Sockets;  using System.IO;  IPEndPoint iPEndPoint = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 80);  TcpListener listener = new  TcpListener(iPEndPoint);  listener.Start();  Socket sk = listener.AcceptSocket();  NetworkStream ns = new  NetworkStream(sk);  StreamReader sr = new StreamReader(ns);  StreamWriter sw = new StreamWriter(ns);  while (true) {  String dl = sr.ReadLine();  if (dl == "bye"){  sw.WriteLine(dl); sw.Flush();  break; }  Console.WriteLine(dl);  sw.WriteLine(dl); sw.Flush();  }  ns.Close(); sr.Close(); sw.Close(); | **Client**  using System.Net;  using System.Net.Sockets;  using System.IO;  Socket sk = new  Socket(AddressFamily.InterNetwork,  SocketType.Stream, ProtocolType.Tcp);  IPEndPoint dcsv = new  IPEndPoint(IPAddress.Parse("127.0.0.1"), 80);  sk.Connect(dcsv);  string input;  while (true){  Console.WriteLine("Nhap lenh: ");  input = Console.ReadLine();  NetworkStream ns = new  NetworkStream(sk);  StreamReader sr = new StreamReader(ns);  StreamWriter sw = new StreamWriter(ns);  sw.WriteLine(input);  sw.Flush();  input = sr.ReadLine();  Console.WriteLine(input);  ns.Close(); sr.Close(); sw.Close(); |

# TCPClient C# chung

|  |  |
| --- | --- |
| **Server**  Socket sk = new  Socket(AddressFamily.InterNetwork,  SocketType.Stream, ProtocolType.Tcp);  IPEndPoint dc = new  IPEndPoint(IPAddress.Any, 80);  sk.Bind(dc); sk.Listen(10);  Console.WriteLine("Waiting...");  Socket client = sk.Accept();  NetworkStream ns = new  NetworkStream(client);  StreamReader sr = new StreamReader(ns);  StreamWriter sw = new StreamWriter(ns);  while (true) { String dl = sr.ReadLine();  Console.WriteLine(dl);  sw.WriteLine(dl);  sw.Flush(); }  sk.Close(); ns.Close();  sr.Close(); sw.Close(); | **Client**  using System.Net; using System.IO;  using System.Net.Sockets;  TcpClient client = new TcpClient();  client.Connect("127.0.0.1", 80);  Stream stream = client.GetStream();  Console.WriteLine("Da ket noi den sver");  while (true) {  Console.WriteLine("Nhap lenh : ");  string c = Console.ReadLine();  StreamReader sr = new  StreamReader(client.GetStream());  StreamWriter sw = new  StreamWriter(client.GetStream());  sw.WriteLine(c); sw.Flush();  c = sr.ReadLine();Console.WriteLine(c);  if (c == "bye"){  sw.WriteLine(c); sw.Flush(); break;}  sr.Close(); sw.Close(); } |

# Tổng hiệu 2 số (python tcp)

|  |  |
| --- | --- |
| **Server:** import socket  addr = (host, port)  sk=socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  sk.bind(addr) sk.listen(5)  print("Đang nghe client")  conn, addr = socket.accept()  data = conn.recv(2048).decode('utf-8')  print("đã nhận kết nối từ client",data)  array = conn.recv(2048).decode('utf-8').split('-')  print(array)  a1 = int(array[0]) a2 = int(array[1])  gt = conn.recv(2048).decode('utf-8')  print("client nhận option: ",gt)  if gt=='1': a3 = a1+a2  print("tổng là : ", a3)  if gt=='2': a3 = a1-a2  print("hiệu là : ", a3)  conn.close() sk.close() | **Client:** import socket  host = 'localhost'  port = 9050  addr = (host, port)  sk=socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  sk.connect(addr)  sk.send(("r").encode('utf-8'))  a = input("mời bạn nhập a: ")  b = input("mời bạn nhập b: ")  m = str(a) + '-'+ str(b)  sk.send(m.encode('utf-8'))  op = input("chọn option:")  sk.send(op.encode('utf-8'))  sk.close() |

# Chuẩn hóa xâu

## TCP

|  |  |
| --- | --- |
| **Server** import socket  sk = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)   sk.bind((host,port)) sk.listen(5)  print("Waiting")  client, addr = sk.accept()  client.send('hello chient'.encode('utf-8'))  data= client.recv(1024)  print(data.decode('utf-8'))  while True:  data= client.recv(1024)  if not data or data.decode('utf-8') =='bye': break  data = data.decode('utf-8')  s = data[0].upper() + data[1:]  snew=''  id=0  for i in s:  if s[id-1] =='.' and s[id+1] != ' ':  snew = snew + ' '+ i.upper()  elif s[id-1] ==',' and s[id] != ' ':  snew = snew + ' ' + i  elif id + 1 == len(s): snew = snew + i  else: snew = snew + i  id=id+1  client.send(snew.encode('utf-8'))client.close() sk.close() | **Client** import socket if \_\_name\_\_ == '\_\_main\_\_':  sk = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  host = 'localhost'  port = 80  addr = (host, port)  sk.connect(addr)  data=sk.recv(1024)  print(data.decode('utf-8'))  s='Hello Server'  sk.send(s.encode('utf-8'))  while True:  c = input('Nhap du lieu: ')  if c == 'bye':  sk.send(c.encode('utf-8'))  break  sk.send(c.encode('utf-8'))  data = sk.recv(1024)  print(data.decode('utf-8'))  sk.close() |

## UDP

|  |  |
| --- | --- |
| **Server**  sk = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)   sk.bind((host,port))   while True:  data,addr= client.recvfrom(1024)  if not data or data.decode('utf-8') =='bye': break  data = data.decode('utf-8')  s = data[0].upper() + data[1:]  snew=''  id=0  for i in s:  if s[id-1] =='.' and s[id+1] != ' ':  snew = snew + ' '+ i.upper()  elif s[id-1] ==',' and s[id] != ' ':  snew = snew + ' ' + i  elif id + 1 == len(s): snew = snew + i  else: snew = snew + i  id=id+1  sk.sendto(snew.encode('utf-8'),addr)sk.close() | **Client**  import socket if \_\_name\_\_ == '\_\_main\_\_':  host = ‘localhost’  port = 80  sk = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)   while True:  c = input('Nhap du lieu: ')  sk.sendto(c.encode('utf-8'),(host,port))  if c == 'bye':  break  data,addr = sk.recvfrom(1024)  print(data.decode('utf-8'))  sk.close() |

# Xdựng ctrình, client yêu cầu user nhập ‘r’ để bđ knoi đến sver. Sau khi svẻ nhận tín hiệu thì trả về các option (1.tru/2.cong). C nhận các option từ sv in ra màn hình. Sau đó yêu cầu user nhập a,b,c. Truyền gtrị cho sv để tính kquả, và trả về tổng hoặc hiệu.

## TCP

|  |  |
| --- | --- |
| **Server**  sk = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)   sk.bind((host, port))  sk.listen(10)  client, addr = sk.accept()  s = '1.tru / 2.cong'  client.send(s.encode('utf-8'))  data = client.recv(1024)  print(data.decode('utf-8'))  while True:  data = client.recv(1024)  print('Client gui:', data.decode('utf-8'))  if data.decode('utf-8') == 'exit': break  else:  try: arr = data.decode('utf-8').split()  if int(arr[0]) == 1:  kq = int(arr[1]) - int(arr[2]) - int(arr[3])  kq = 'Hieu 3 so la:' + str(kq)  client.send(kq.encode('utf-8'))  elif int(arr[0]) == 2:  kq = int(arr[1]) + int(arr[2]) + int(arr[3])  kq = 'Tong 3 so la:' + str(kq)  client.send(kq.encode('utf-8'))  else: client.send(data)  except: client.send(data)  client.close() sk.close() | **Client**  import socket if \_\_name\_\_ == '\_\_main\_\_':  sk = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  s = input('Nhap r de ket noi:')  if s == 'r':  host = '127.0.0.1'  port = 1999  sk.connect((host, port))  data = sk.recv(1024)  print(data.decode('utf-8'))  while True:  c = input('Nhap option va 3 so:')  sk.send(c.encode('utf-8'))  if c == 'exit':  break  else:  data = sk.recv(1024)  print('server gui:', data.decode('utf-8'))  sk.close()  else:  print('Khong ket noi duoc') |

## UDP

|  |  |
| --- | --- |
| **Server**  sk = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)   sk.bind((host, port))  s = '1.tru / 2.cong'  data, addr = sk.recvfrom(1024)  sk.sendto(s.encode('utf-8'), addr)  while True:  data, addr = sk.recvfrom(1024)  print('Client gui:', data.decode('utf-8'))  if data.decode('utf-8') == 'exit': break  else:  try: arr = data.decode('utf-8').split()  if int(arr[0]) == 1:  kq = int(arr[1]) - int(arr[2]) - int(arr[3])  kq = 'Hieu 3 so la:' + str(kq)  sk.sendto(kq.encode('utf-8'), addr)  elif int(arr[0]) == 2:  kq = int(arr[1]) + int(arr[2]) + int(arr[3])  kq = 'Tong 3 so la:' + str(kq)  sk.sendto(kq.encode('utf-8'), addr)  else: sk.sendto(data, addr)  except: sk.sendto(data, addr)  sk.close() | **Client**  import socket if \_\_name\_\_ == '\_\_main\_\_':  sk = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  s = input('Nhap 'r' de ket noi:')  if s == 'r':  host = '127.0.0.1'  port = 9999  sk.sendto(s.encode('utf-8'), (host, port))  data, addr = sk.recvfrom(1024)  print(data.decode('utf-8'))  while True:  s = input('Chon opt va nhap 3 so:')  sk.sendto(s.encode('utf-8'), (host, port))  if s == 'exit':  break  else:  data, addr = sk.recvfrom(1024)  print('server gui:', data.decode('utf-8'))  sk.close()  else:  print('Khong ket noi duoc') |

## C# TCP

|  |  |
| --- | --- |
| **Server**  Socket socket = new  Socket(AddressFamily.InterNetwork,  SocketType.Stream, ProtocolType.Tcp);  IPEndPoint ipEndPoint = new  IPEndPoint(IPAddress.Any, 9050);  socket.Bind(ipEndPoint);  socket.Listen(10);  Console.WriteLine("waiting...");  Socket client = socket.Accept();  Console.WriteLine("Connect to: "+client.RemoteEndPoint);  NetworkStream ns = new  NetworkStream(client);  StreamReader sr = new StreamReader(ns); StreamWriter sw = new StreamWriter(ns);  sw.WriteLine("1.tru/2.cong"); sw.Flush();  while (true) {  var option = sr.ReadLine();  if (option == "bye" || option == null)  { break; }  try{  var arr = option.Split(' ');  if (arr[0] == "1"){  var hieu = int.Parse(arr[1]) - int.Parse(arr[2]) - int.Parse(arr[3]);  sw.WriteLine("Hieu: "+hieu);  sw.Flush();}  else if (arr[0] == "2"){  var tong = int.Parse(arr[1]) + int.Parse(arr[2]) + int.Parse(arr[3]);  sw.WriteLine("Tong: "+tong);  sw.Flush();}  else{  sw.WriteLine("option nhap sai");  sw.Flush();}  }catch (Exception e) {  sw.WriteLine("Du lieu nhap sai");  sw.Flush();}  }  socket.Close(); client.Close();  Console.ReadLine(); | **Client**  using System.IO;  using System.Net;  using System.Net.Sockets;  Socket server = new  Socket(AddressFamily.InterNetwork,  SocketType.Stream, ProtocolType.Tcp);  IPEndPoint ipEndPoint = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 9050);  Console.WriteLine("Nhap r de ket noi: ");  var r = Console.ReadLine();  while (r != "r"){  Console.WriteLine("Nhap r de ket noi: ");  r = Console.ReadLine();}  server.Connect(ipEndPoint);  NetworkStream ns = new  NetworkStream(server);  StreamReader sr = new StreamReader(ns);  StreamWriter sw = new StreamWriter(ns);  var data = sr.ReadLine();  Console.WriteLine(data);  while (true) {  Console.WriteLine("Nhap option");  var option = Console.ReadLine();  if (option == "bye"|| option == null) {  sw.WriteLine(option);  sw.Flush();  break; }  Console.WriteLine("Nhap a: ");  var a = Console.ReadLine();  Console.WriteLine("Nhap b: ");  var b = Console.ReadLine();  Console.WriteLine("Nhap c: ");  var c = Console.ReadLine();  var send\_data = option + " " + a + " " + b + " " + c;  sw.WriteLine(send\_data);  sw.Flush();  var result = sr.ReadLine();  Console.WriteLine(result); }  server.Close();  Console.ReadLine(); |

## UDP C#

|  |  |
| --- | --- |
| **Server**  byte[] dl = new byte[1024]; int kt;  Socket socket = new  Socket(AddressFamily.InterNetwork,  SocketType.Dgram, ProtocolType.Udp);  IPEndPoint ipEndPoint = new  IPEndPoint(IPAddress.Any, 9050);  socket.Bind(ipEndPoint);  Console.WriteLine("waiting...");  IPEndPoint client = new  IPEndPoint(IPAddress.Any, 0);  EndPoint remote = (EndPoint) client;  string data = "";  kt = socket.ReceiveFrom(dl, ref remote);  data = Encoding.ASCII.GetString(dl, 0, kt); Console.WriteLine(data);  data = "1.tru/2.cong";  dl = Encoding.ASCII.GetBytes(data);  socket.SendTo(dl, dl.Length,  SocketFlags.None, remote);  while (true){  dl = new byte[1024];  kt = socket.ReceiveFrom(dl, ref remote);  string option =  Encoding.ASCII.GetString(dl, 0, kt);  if (option == "bye" || option == null)  { break; }  try{ var arr = option.Split(' ');  if (arr[0] == "1"){  int hieu = int.Parse(arr[1]) - int.Parse(arr[2]) - int.Parse(arr[3]);  string send\_data = hieu.ToString();  dl = Encoding.ASCII.GetBytes(send\_data);  socket.SendTo(dl, remote); }  else if (arr[0] == "2"){  int tong = int.Parse(arr[1]) + int.Parse(arr[2]) + int.Parse(arr[3]);  string send\_data =tong.ToString();  dl = Encoding.ASCII.GetBytes(send\_data);  socket.SendTo(dl, remote); }  else{ dl = Encoding.ASCII.GetBytes("option nhap sai");  socket.SendTo(dl, remote); }  }catch (Exception e) {  dl = new byte[1024];  dl = Encoding.ASCII.GetBytes("Du lieu nhan sai");  socket.SendTo(dl, remote); }  } socket.Close(); | **Client**  using System.Net;  using System.Net.Sockets;  Socket socket = new  Socket(AddressFamily.InterNetwork,  SocketType.Dgram, ProtocolType.Udp);  IPEndPoint ipEndPoint = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 9050);  byte[] dl = new byte[1024];  int kt; string data = "hello server";  dl = Encoding.ASCII.GetBytes(data);  socket.SendTo(dl, dl.Length,  SocketFlags.None, ipEndPoint);  IPEndPoint server = new  IPEndPoint(IPAddress.Any, 0);  EndPoint remote = (EndPoint)server;  Console.WriteLine("Nhap r de bd: ");  string r = Console.ReadLine();  while (r != "r"){  Console.WriteLine("Nhap r de bd: ");  r = Console.ReadLine();}  kt = socket.ReceiveFrom(dl, ref remote);  data = Encoding.ASCII.GetString(dl, 0, kt);  Console.WriteLine(data);  while (true) {  dl = new byte[1024];  Console.WriteLine("Nhap option");  string option = Console.ReadLine();  if (option == "bye" || option == null) {  dl =  Encoding.ASCII.GetBytes(option);  socket.SendTo(dl, remote);  break; }  Console.WriteLine("Nhap a: ");  string a = Console.ReadLine();  Console.WriteLine("Nhap b: ");  string b = Console.ReadLine();  Console.WriteLine("Nhap c: ");  string c = Console.ReadLine();  string send\_data = option + " " + a + " " + b + " " + c;  dl =  Encoding.ASCII.GetBytes(send\_data);  socket.SendTo(dl, remote);  dl = new byte[1024];  kt = socket.ReceiveFrom(dl, ref remote);  data = Encoding.ASCII.GetString(dl, 0, kt);  Console.WriteLine(data); }  socket.Close(); |

# Sver lưu dssv gồm: msv,tên sv. Client gửi lời chào sver-sver gửi lại. Sver gửi thông điệp ‘nhập 1 để bđ’ if client gửi 1 thì server yêu cầu nhập msv. Sver nhận msv và trả về thông tin tương ứng với msv

## TCP

|  |  |
| --- | --- |
| **Server**  sk = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)   sk.bind((host, port)) sk.listen(10)  client, addr = sk.accept()  arr1 = ['001', '002', '003', '004']  arr2 = ['HA', 'MT', 'HV', 'HG']   print(client.recv(1024).decode('utf-8'))   client.send('Hello client'.encode('utf-8'))  client.send('Nhap 1 de bd'.encode('utf-8'))  a = client.recv(1024)  if a.decode('utf-8') == '1':  while True:  client.send('Nhap msv'.encode('utf-8'))   print(client.recv(1024).decode('utf-8'))  if c.decode('utf-8') =='bye': break  else: ktra = False  for i in range(0, len(arr1)):  if arr1[i] == c.decode('utf-8'):  ktra = True  s = 'Ten sinh vien: ' + arr2[i]  client.send(s.encode('utf-8'))  print(s)  break  if ktra == False:  client.send('Khong tim thay sinh vien'.encode('utf-8'))  else: client.send(a)  client.close() sk.close() | **Client**  import socket if \_\_name\_\_ == '\_\_main\_\_':  sk=socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  sk.connect(('127.0.0.1', 6789))  s = 'Hello server'  sk.send(s.encode('utf-8'))  s = sk.recv(1024)  print(s.decode('utf-8'))  a= sk.recv(1024)   s = input(a.decode('utf-8'))  sk.send(s.encode('utf-8'))  if s == '1':  while True:  a = sk.recv(1024)   z = input(a.decode('utf-8'))  sk.send(z.encode('utf-8'))  if z==’bye’: break  else:  print(sk.recv(1024).decode('utf-8'))  else:  print(sk.recv(1024).decode('utf-8'))  sk.close() |

## UDP

|  |  |
| --- | --- |
| **Server**  arr1 = ['1', '2', '3']  arr2 = ['HA', 'HN', 'VN']  sk = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  sk.bind(('localhost', 2411))  data, addr = sk.recvfrom(1024)  print("client gui: ", data.decode('utf-8'))  s = 'hello client'  sk.sendto(s.encode('utf-8'), addr)  s = 'nhap 1 de bat dau: '  sk.sendto(s.encode('utf-8'), addr)  data, addr = sk.recvfrom(1024)  print("client gui: ", data.decode('utf-8'))  if data.decode('utf-8') == '1':  while True:  s = 'nhap ma sinh vien: '  sk.sendto(s.encode('utf-8'), addr)  data, addr = sk.recvfrom(1024)  print("client gui: ", data.decode('utf-8'))  if data.decode('utf-8') =='bye': break  else:  ktra = False  for i in range(0, len(arr1)):  if arr1[i] == data.decode('utf-8'):  ktra = True  sk.sendto(arr2[i].encode('utf-8'), addr)  break  if ktra == False:  sk.sendto('Khong co sinh vien'.encode('utf-8'), addr)  else: sk.sendto(data,addr)  sk.close() | **Client**  import socket  if \_\_name\_\_ == '\_\_main\_\_':  sk = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  host = '127.0.0.1'  port = 2411  s = 'Hello server'  sk.sendto(s.encode('utf-8'), (host, port))  data, addr = sk.recvfrom(1024)  print('Server gui:', data.decode('utf-8'))  data, addr = sk.recvfrom(1024)  print(data.decode('utf-8'))  s = input(data.decode('utf-8'))  sk.sendto(s.encode('utf-8'), (host, port))  if s == '1':  while True:  data, addr = sk.recvfrom(1024)   s = input(data.decode('utf-8'))  sk.sendto(s.encode('utf-8'), (host, port))  if s == 'bye': break  else:  data, addr = sk.recvfrom(1024)  print(data.decode('utf-8'))  else:  data, addr = sk.recvfrom(1024)  print(data.decode('utf-8'))  sk.close() |

# Client gửi UP(LO) và 1 xâu ký tự đến sver. Và nhận kq trả về. Sver hiện thông tin client, ktra lệnh nhận đc là gì? Nhận xâu. Thực hiện biến đổi xâu thành chữ hoa (UP) or thường(LO). Trả kq về client.

## TCP

|  |  |
| --- | --- |
| **Server**  sk = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)   sk.bind((host, port))  sk.listen(10)  client, addr = sk.accept()  print('Dia chi client: {}',format(addr))  while True: data = client.recv(1024)  print("client gui:",data.decode('utf-8'))  arr = data.decode('utf-8').split()  if arr[0] == 'UP': s = ''  for i in range(1, len(arr)):  s = s + arr[i].upper() + ' '  print(s)  client.send(s.encode('utf-8'))  elif arr[0] == 'LO': s = ''  for i in range(1, len(arr)):  s = s + arr[i].lower() + ' '  print(s)  client.send(s.encode('utf-8'))  else:  client.send('Khong dung lenh’.encode('utf-8'))  client.close() sk.close() | **Client**  import socket  if \_\_name\_\_ == '\_\_main\_\_':  sk = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)   host = 'localhost'  port = 1999  sk.connect((host, port))   while True:  s = input('Nhap UP hoac LO va 1 xau ki tu:')  sk.send(s.encode('utf-8'))  print(sk.recv(1024).decode('utf-8'))  sk.close() |

## UDP

|  |  |
| --- | --- |
| **Server**  sk = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)   sk.bind((host, port))   while True:  data, addr = sk.recvfrom(1024)  print("client gui:", data.decode('utf-8'))  arr = data.decode('utf-8').split()  if arr[0] == 'UP': s = ''  for i in range(1, len(arr)):  s = s + arr[i].upper() + ' '  print(s)  sk.sendto(s.encode('utf-8'),addr)  elif arr[0] == 'LO': s = ''  for i in range(1, len(arr)):  s = s + arr[i].lower() + ' '  print(s)  sk.sendto(s.encode('utf-8'),addr)  else: sk.sendto('Khong dung lenh'.encode('utf-8'),addr)  sk.close() | **Client**  import socket if \_\_name\_\_ == '\_\_main\_\_':  sk = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  while True:  s = input('Nhap UP hoac LO va 1 xau ki tu:')  sk.sendto(s.encode('utf-8'), (host,port))  data, addr = sk.recvfrom(1024)  print('Server gui:', data.decode('utf-8'))  sk.close()  else:  print('Khong ket noi voi server') |

## 

## TCP C#

|  |  |
| --- | --- |
| **Server**  Socket sk = new  Socket(AddressFamily.InterNetwork,  SocketType.Stream, ProtocolType.Tcp);  IPEndPoint ipEndPoint = new  IPEndPoint(IPAddress.Any, 9050);  sk.Bind(ipEndPoint); sk.Listen(10);  Console.WriteLine("Waiting ...");  Socket client = sk.Accept();  Console.WriteLine("Ket noi tu dia chi: " + client.RemoteEndPoint);  NetworkStream ns = new  NetworkStream(client);  StreamReader sr = new StreamReader(ns);  StreamWriter sw = new StreamWriter(ns);  var data = sr.ReadLine();  Console.WriteLine(“Client : " + data);  sw.WriteLine("Hello client"); sw.Flush();  while (true) {  var option = sr.ReadLine();  if (option == "bye" || option == null){  Console.WriteLine("Thoat"); break; }  var arr = option.Split(' ');  if (arr[0] == "UP" || arr[0] == "Up" || arr[0]=="up" || arr[0]=="uP"){  var data\_send = "";  for (int i = 1; i < arr.Length; i++) {  data\_send += arr[i].ToUpper()+ " ";}  Console.WriteLine(data\_send);  sw.WriteLine(data\_send);  sw.Flush();}  else if (arr[0] == "LO" || arr[0] == "Lo" || arr[0] == "lo" || arr[0] == "lO"){  var data\_send = "";  for(int i = 1; i < arr.Length; i++){  data\_send += arr[i].ToLower() + " ";}  Console.WriteLine(data\_send);  sw.WriteLine(data\_send);  sw.Flush();}  else {sw.WriteLine("Nhap sai lenh");  sw.Flush();}  }  client.Close(); sk.Close(); Console.ReadLine(); | **Client**  using System.IO;  using System.Net;  using System.Net.Sockets;  Socket server = new  Socket(AddressFamily.InterNetwork,  SocketType.Stream, ProtocolType.Tcp);  IPEndPoint iPEndPoint = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 9050);  server.Connect(iPEndPoint);  NetworkStream ns = new  NetworkStream(server);  StreamReader sr = new StreamReader(ns);  StreamWriter sw = new StreamWriter(ns);  sw.WriteLine("Hello server"); sw.Flush();  var data = sr.ReadLine();  Console.WriteLine("Du lieu nhan tu server: " + data);  while (true) {  Console.WriteLine("Nhap lenh: ");  var option = Console.ReadLine();  if (option == "bye" || option == null) {  sw.WriteLine(option); sw.Flush();  Console.WriteLine("Thoat");  break; }  else{  Console.WriteLine("Nhap c ki tu:");  var kitu = Console.ReadLine();  var data\_send = option +" "+ kitu;  sw.WriteLine(data\_send);  sw.Flush();}  var result = sr.ReadLine();  Console.WriteLine(result);  }  Console.ReadLine();  server.Close(); |

# Chia UDP

|  |  |
| --- | --- |
| **Server**  import socket  sk = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  sk.bind(('localhost', 2411))  data, addr = sk.recvfrom(1024)  print("client gui: ", data.decode('utf-8')) sk.sendto('hello client'.encode('utf-8'),addr)  while True:  data, addr = sk.recvfrom(1024)  print("client gui:",data.decode('utf-8'))  if data.decode('utf-8') == "exit": break  else:  try:  arr = data.decode('utf-8').split()  so1 = int(arr[0])  so2 = int(arr[1])  if so2 == 0:  print("khong chia duoc")  sk.sendto(data, addr)  else:  kq = so1 / so2  kq = str(kq)  sk.sendto(kq.encode('utf-8'), addr)  except:  print("client khong gui so")  sk.sendto(data, addr)  sk.close() | **Client**  import socket  if \_\_name\_\_ == '\_\_main\_\_':  sk = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  s = 'hello server'  sk.sendto(s.encode('utf-8'), ('localhost', 2411))  data, addr = sk.recvfrom(1024)  print("server gui: ", data.decode('utf-8'))  while True:  c = input("Nhap lenh: ")  sk.sendto(c.encode('utf-8'), ('localhost', 2411))  if c == "exit":  print("exit")  break  else:  data, addr = sk.recvfrom(1024)  print("server gui: ", data.decode('utf-8'))  sk.close() |

# Lý thuyết:

1. Địa chỉ IP là địa chỉ của 1 thiết bị trên mạng, dùng để phân biệt các máy tính với nhau trên mạng Internet.

2. Port (cổng): là các số hiệu địa chỉ giúp phân biệt dữ liệu nào thuộc ứng dụng nào: được chèn vào phần đầu của gói tin và được định nghĩa ở tầng giao vận.

3. Địa chỉ IP là một số nguyên 32 bit chia thành 4 byte: được ngăn cách với nhau bởi dấu chấm; chia làm hai phần: địa chỉ mạng và địa chỉ máy

4. IP cục bộ là địa chỉ ảo of máy tính of bạn trong mạng of cty hay mạng ở nhà riêng. IP này chỉ các máy trong mạng biết và trao đổi với nhau bằng IP này. Tùy vào cấu hình của modem mạng sẽ tự động cấp phát IP cho các máy con trong mạng như mtính để bàn, ltop,.

5. IP toàn của là IP của bạn khi truy cập vào mạng Internet, truy cập trang web hay gửi mail thì IP của bạn sẽ bị máy chủ của dịch vụ web hay email lưu lại.

6. IP động là IP không cố định, nó sẽ được thay đổi khi một số điều kiện nhất định xảy ra.

VD: khi khởi động một modem Internet sẽ được nhà mạng cấp một địac hỉ IP mới.

7. IP tĩnh là Ip cố định, không thay đổi khi khởi động modem.

8. Chuẩn Internet: đảm bảo các ứng dụng mạng hoạt động trên các môi trường ≠ nhau.

9. Namespace là 1 gói các thực thể có các thuộc tính & phương thức độc lập vs bên ngoài.

10. Socket là 1 giao diện lập trình ứng dụng mạng. Qua giao diện này, ta có thể lập trình điều khiển việc truyền thông giữa 2 máy sử dụng giao thức TCP, UDP. 2 loại socket: STREAM và DATAGRAM

11. Phương thức hoạt động của UDP: +Gói tin chỉ gửi đến bên nhận. + Bên gửi sẽ ko chờ đợi để chắc chắn rằng bên nhận đã nhận được các gói tin, nó sẽ tiếp tục gửi các gói tin tiếp theo. + Khi 1 số ứng dụng dựa trên giao thức UDP gửi dữ liệu đến 1 host ≠ trên mạng, UDP thêm vào 1 header có độ dài 8 byte chứa các số hiệu cổng nguồn và đích, cùng với tổng chiều dài dữ liệu và thông tin checksum. IP thêm vào header của riêng nó vào đầu mỗi datagram UDP để tạo nên 1 datagram IP.

12. Nhược điểm của UDP: Thiếu các tín hiệu bắt tay