Networking

Th.S Trần Đức Lợi
Pythonvietnam.info

Ôn tập bài cũ

- Ôn lại nội dung đã học về **Thread**

Mục đích bài học

- Tìm hiểu về network trong python:
  - Network basics
  - TCP/IP Client and Server
  - User Datagram Client and Server

Network basics: Socket

- Socket là gì?
  - Socket là điểm đầu cuối của một kênh giao tiếp sử dụng bởi chương trình để truyền dữ liệu qua lại
- Một socket có 2 thuộc tính:
  - Address family
  - Socket type

Network basics: Address Family

- Python hỗ trợ 3 họ địa chỉ:
  - **AF_INET** (đánh địa chỉ IP v4)
  - AF_INET6 (đánh địa chỉ IP v6)
  - AF_UNIX (đánh địa chỉ UDS – Unix Domain Socket)

Network basics: Socket types

- Có 2 loại socket type thông thường hay sử dụng:
  - SOCK_DGRAM (UDP)
  - SOCK_STREAM (TCP)
- UDP vs TCP?

Network basics: Sử dụng socket

- import **socket**
- **print** socket.gethostname()
  - Trả về tên chính thức của host hiện tại
- Print socket.gethostbyname(host)
  - Trả về IP sau khi phân giải xong host

Network basics: Sử dụng socket

- hostname, aliases, addresses = socket.gethostbyname_ex(host)
  - Để lấy được nhiều thông tin hơn về host
- Sử dụng socket để reverse lookup:
  - hostname, aliases, addresses = socket.gethostbyaddr('172.0.0.7')

Network basics

- Tại một thời điểm chỉ có 1 socket được sử dụng 1 địa chỉ IP, 1 port và 1 giao thức
- 3 thông số này định nghĩa ra 1 kênh giao tiếp
- Một vài port đã được chỉ định sẵn cho 1 số giao thức
  - http: 80
  - https: 443
  - ftp: 21

Network basics: Sử dụng socket

- Có thể dùng python để tra cứu cổng chuẩn từ đường link giao thức:
  - Smtp://mail.google.com -> 25
- Sử dụng hàm **getservbyname()** của thư viện socket
  - Port = socket.getservbyname(urlparse. Urlparse(link).scheme)

TCP/IP Server

- Socket thường được dùng theo mô hình server-client
- Một ứng dụng đóng vai tròng server
- Các ứng dụng khác đóng vai trò client
- Giao tiếp là giao tiếp 2 chiều

Server

- Quá trình dựng server với python:
  - Khai báo một socket
  - Bind socket này vào địa chỉ của server
  - Bắt đầu lắng nghe trên địa chỉ đã bind
  - Tiếp nhận và accept kết nối
  - Xử lý dữ liệu, trả về kết quả cho client
  - Đóng kết nối

Server

- Tạo một socket
  - sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
- Bind socket này vào địa chỉ của server:
  - sock.bind((‘ip’, port))

Server

- Lắng nghe kết nối (socket đã ở chế độ server):
  - Sock.listen(1)
- Chấp nhận kết nối:
  - connection, client_address = sock.accept()
  - Connection: đại diện kết nối từ server đến client
  - Client_address: địa chỉ client
  - Connection: thực chất là 1 socket khác với số port khác được HĐH tự phân bố

Server

- Nhận dữ liệu gửi lên từ client:
  - Connection.recv(số bytes)
- Truyền dữ liệu đến client:
  - Connection.sendall()

Server

- Đóng kết nối tới client:
  - Connection.close()

Client

- Quy trình xây dựng một socket client:
  - Tạo một socket mới
  - Dùng socket vừa tạo, kết nối tới địa chỉ của server(bao gồm IP và port)
  - Truyền và nhận dữ liệu với server
  - Đóng socket

Client

- Tạo một socket mới:
  - sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
- Kết nối tới server:
  - sock.connect((‘IP’, port))

Client

- Gửi dữ liệu lên server:
  - Sock.sendall()
- Nhận dữ liệu từ server gửi về:
  - Sock.recv(số bytes)

Client

- Đóng socket:
  - Socket.close()

Bài tập

- Xây dựng một server mà sẽ thực hiện trả về đảo ngược chuỗi truyền lên từ client

Bài tập

- Xây dựng ứng dụng chat console giữa hai máy tính trong mạng LAN

Bài tập

- Xây dựng ứng dụng chat console giữa 2 máy tính trong mạng LAN sử dụng đa luồng để xử lý dữ liệu chat. (YM, Zalo, FB, …)

Tổng kết bài học

- Networking basics
- TCP Server
- TCP Client