

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CÁ NHÂN LẬP TRÌNH SOCKET**

**Sinh viên thực hiện: 19120557 - Trần Tuấn Kiệt**

**Giảng viên hướng dẫn:**

**Lê Hà Minh  
Nguyễn Thanh Quân  
Chung Thủy Linh**

CHẤM ĐIỂM	
Bảng số	Bảng chữ

**TP. HỒ CHÍ MINH - NĂM 2021**

## MỤC LỤC

Bảng phân công công việc.....	2
Hướng dẫn sử dụng.....	3
Kịch bản giao tiếp.....	7
Kiểu dữ liệu của thông điệp.....	10
Cách tổ chức cơ sở dữ liệu .....	10
Các hàm xử lý.....	10
Nguồn tham khảo.....	12

### Nội dung công việc

Công việc	Mô tả
Kết nối	TCP, cho phép kết nối nhiều clients chung một server. Nhập IP server để kết nối.
Quản lý kết nối:	tránh báo lỗi khi user thoát không đúng cách. (tránh để cho server bị treo)
Dùng Thread	để điều phối nhiều chương trình cùng một lúc
Đăng nhập và đăng kí	Chỉ cho phép client sử dụng các chức năng của server khi đã đăng nhập Phát hiện đã tồn tại tài khoản trùng khi đăng kí
Tra cứu	Đọc dữ liệu trong database và đọc dữ liệu này lên. User nhập date và type (ngày và loại vàng). Server sẽ trả về kết quả dựa trên 2 thông tin này
Quản lí cơ sở dữ liệu	Gửi request và lấy thông tin. Lưu các thông tin này xuống file có cấu trúc JSON
Ngắt kết nối	Client gửi thông báo ngắt kết nối. Server được ngắt kết nối đến tất cả users.
Giao diện	giao diện trên CMD đơn giản, xử lí các thông báo gửi từ server đến client và in ra màn hình client. Không làm GUI

### Hướng dẫn sử dụng:

**Bước 1:** Tải 2 file để thực thi chương trình, 1 file cho server và 1 file cho client. Đi kèm với đó là file “user\_management.txt” chứa sẵn account tạo sẵn. Sau này có thể dùng account này để đăng nhập, hoặc nếu không ta có thể tạo mới.

**Bước 2 - Chạy chương trình:** Khởi động chương trình nào trước cũng được. Khởi động bằng cách nhấn F5 (visual studio 2019) hoặc dùng lệnh từ command prompt: “python server.py” và “python client.py”

(bên trái: server.py, bên phải client.py)

Lần đầu tiên chạy “*python server.py*” ta cần cấp phát quyền truy cập Firewall nếu có thông báo hiện ra.

**Bước 3** - Cập nhập database: server tự động cập nhập dữ liệu request từ API do giảng viên cung cấp, tự động mỗi 30 phút. Client không can thiệp vào quá trình này.

```
Server is waiting for orders
~SERVER~ Updating database
~SERVER~ Database is up-to-date
```

**Bước 4:** tại màn hình CMD client, ta nhập địa chỉ IP cần kết nối. (lưu ý: địa chỉ IP này khác nhau nếu server chạy trên máy khác nhau). Trong trường hợp cả server và client đều chạy chung trên 1 máy, ta sẽ nhập “myip” để chương trình tự động điền IP của máy đang chạy. Nếu không tìm thấy server IP, ta buộc phải nhập lại. Nhập đúng sẽ chuyển sang bước tiếp theo

```
Input SERVER IP to continue:
Or type "myip" to input your IP (In case you are running both on the same computer)
>192.111.66.6

CONNECTION FAILED
Input SERVER IP to continue:
Or type "myip" to input your IP (In case you are running both on the same computer)
>
```

(Nhập sai IP: 192.111.66.6, chương trình sẽ mất một khoảng thời gian ngắn-30s trước khi báo lỗi kết nối: **CONNECTION FAILED**)

```

CONNECTION FAILED
Input SERVER IP to continue:
Or type "myip" to input your IP (In case you are running both on the same computer)
>192.168.23.1

SOCKET PROGRAMING
SERVER NAME: mothership
SERVER IP: 192.168.23.1

Đề 1: Tỷ Giá Vàng Việt Nam

Authors: 19120557 - Trần Tuấn Kiệt
         20120465 - Hà Thị Hương Giang
         20120479 - Trần Lê Hiếu

Important key words:
-quit: Exit program
-signin: Sign in
-signup: Sign up

CONNECT TO SERVER SUCCESSFULLY
>>_

```

(Nhập đúng IP: 192.168.23.1, program chào đón ta qua màn hình client)

```

SERVER Database is up to date
~SERVER~ DETECTING A NEW CONNECTION FROM: ('192.168.23.1', 55719) <<<

```

(server)

**Bước 5 – Ghi nhớ “key words” quan trọng**, có 3 keyword bắt đầu bằng “-“

- quit: Gửi thông báo đến server để server ngắt kết nối.
- signin: đăng nhập với tài khoản có sẵn.
- signup: đăng kí thông tin lên server.

**Bước 6 – Sign up:**

```

>>-signup

SIGN UP
Type your username and your password (separate with a space " ")
For example: "MyName01 123456"
>>_

```

Ta nhập username và password trên cùng một dòng, cách nhau bởi khoảng trắng.

**Quy tắc:** Username không được phép chứa khoảng trắng. Password được thì được phép.

Nếu thực hiện bước này, ta có thể bỏ qua bước 7.

```

SIGN UP
Type your username and your password (separate with a space " ")
For example: "MyName01 123456"
>>oop dsa
A new account has been created. WELCOME "oop"

```

(client)

```

~('192.168.23.1', 55719)~ is creating new account
~('192.168.23.1', 55719)~ Account created with name: "oop"

```

(server)

Nếu account đã tồn tại, buộc ta phải tạo tài khoản mới hoặc đăng nhập -signin:

```

>>khtn daihoc tphcm
Account is already exists, you can type "-signin" to continue

```

(client)

```

~('192.168.23.1', 58477)~ Account is already exists

```

(server)

**Bước 7 – Sign in:** Trong trường hợp -signin thay vì -signup, ta cũng nhập tương tự như bước 6 (username và password trên cùng một dòng, cách nhau bởi khoảng trắng).

```

>>-signin
SIGN IN
Type your username and your password (separate with a space " ")
For example: "MyName01 123456"
Or you can create new account by typing: "-signup"
>>

```

Ta nhập thử: “khtn daihoc tphcm” với khtn là username và password là “daihoc tphcm”

```

>>khtn daihoc tphcm
Log in successfully! WELCOME "khtn"

```

Màn hình server:

```

~('192.168.23.1', 55719)~ is signing in
~('192.168.23.1', 55719)~ Successfully access: 192.168.23.1 is "khtn"

```

**Bước 8 – Tra cứu:** Chỉ khi màn hình tra cứu xuất hiện. Ta có thể thực hiện bước tiếp theo

```

LOOK UP
Type date and type (separate with a space " ")
For example: "2021-12-27 SCJ"
>>

```

(client)

Xuất ra màn hình client về số kết quả tìm thấy, kết quả chi tiết gồm: **BUY** (mua), **SELL** (bán), **Company** (Công ty), **Brand** (Nhãn hàng) và **ID**. Đóng khung và in hoa **BUY** và **SELL** để làm thông tin quan trọng thu hút người xem

```
12 results found:
|BUY: 60,400.00|
|SELL: 61,800.00|
|Company: 3BANKS|
|Brand: MARITIME BANK|
|ID: 16922864|
|
|BUY: 55,600.00|
|SELL: 56,600.00|
|Company: 3BANKS|
|Brand: SACOMBANK|
|ID: 16940187|
|
|BUY: 61,100.00|
|SELL: 61,450.00|
|Company: 2GROUP|
|Brand: Mi Hồng SJC|
|ID: 16939735|
|
|BUY: 1,811.80|
~('192.168.23.1', 60991)~ is looking up ['2021-12-27', 'SJC']
~('192.168.23.1', 60991)~ 12 results found
```

(client)

(server)

**Cú pháp:** Gõ lệnh trên một dòng: *date* + ' ' + *type*

Trong đó: date (yyyy-mm-dd) là ngày

type: loại vàng mà ta muốn tra cứu (có thể viết khoảng trắng)

**Bước 9 – Ngắt kết nối:** Có 3 cách:

**Cách 1:** Gõ “-quit” từ client để yêu cầu server ngắt kết nối

```
>>-quit
REMOVING CONNECTION
CONNECTION REMOVED. THANK YOU FOR USING OUR PROGRAM
Press any key to continue . . .
```

(client)

```
~('192.168.23.1', 58477)~ has been removed.
~SERVER~ CONNECTION FROM ('192.168.23.1', 58477) HAS BEEN REMOVED<<<
```

(server)

**Cách 2:** Buộc đóng cửa sổ console. Lúc này server sẽ báo mất kết nối và tiến hành loại bỏ kết nối này ngay sau đó (nếu không làm bước này, thì ta không thể kết nối lại sau đó vì đã kết nối này tồn tại kết nối trước đó):

```
~SERVER~ Lost connection to client ('192.168.23.1', 55719)!!!
~SERVER~ CONNECTION FROM ('192.168.23.1', 55719) HAS BEEN REMOVED<<<
```

**Cách 3:** Yêu cầu server ngắt kết nối đến tất cả client bằng cách gõ lệnh “-quit” trong CMD server (chú ý không để nhầm lẫn với Client của cách 1)

```
-quit
~SERVER~ server has seen a command to close all connections.
~SERVER~ Lost connection to client ('192.168.23.1', 52786)!!!
~SERVER~ CONNECTION FROM ('192.168.23.1', 52786) HAS BEEN REMOVED<<<
```

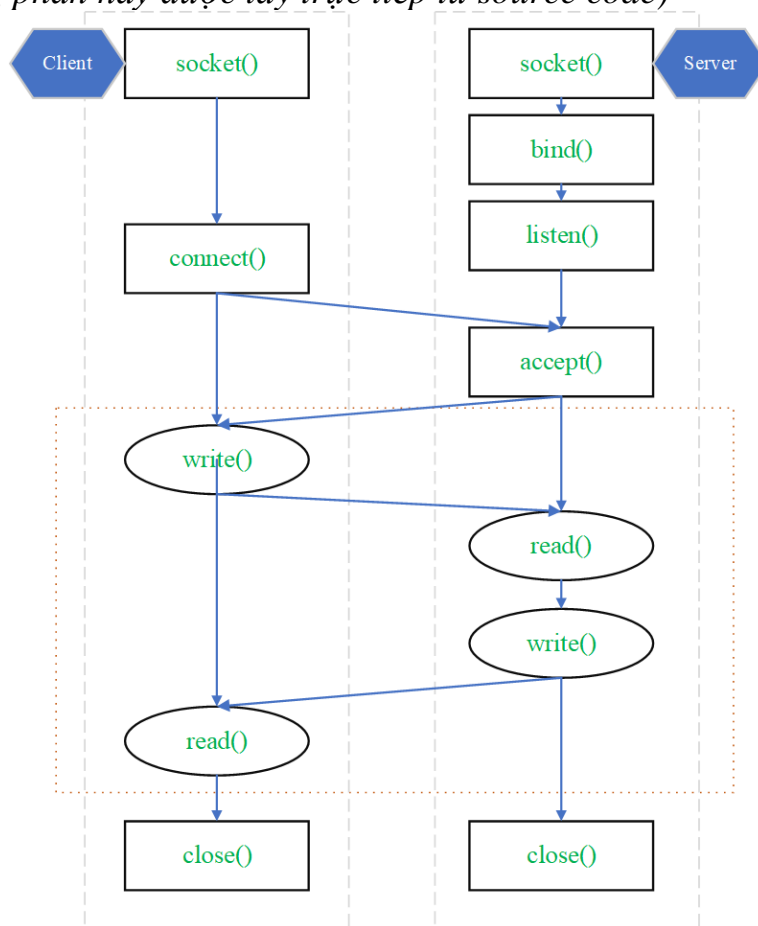
(server)

```
SERVER WAS FORCED TO REMOVED YOUR CONNECTION.
You may close this program
Press any key to continue . . .
```

(client)

### Kịch bản giao tiếp:

(các hàm trong phần này được lấy trực tiếp từ source code)



(Tham khảo từ tài liệu cung cấp bởi giảng viên)

**Giai đoạn 1:** Server tạo một Socket để có thể sử dụng các dịch vụ của tầng vận chuyển:

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

**AF\_INET:** address-family ipv4 (tương tự ta cũng có AF\_INET6: ipv6)

**SOCK\_STREAM**: connection-oriented TCP protocol

Gán số hiệu cổng: `ADDR = (SERVER, PORT)` server được điền thủ công (không hay do có thể chạy program ở máy khác dẫn đến lỗi) hoặc được lấy bằng cách:

```
SERVER = socket.gethostbyname(socket.gethostname())
```

*(socket.gethostname(): trả về tên thiết bị;  
socket.gethostbyname() trả về địa chỉ IP ở dạng str)*

Ta sẽ sử dụng `PORT = 8888` (chọn số hiệu cổng này vì khả năng cao PORT này còn rảnh)

Mỗi khi tạo socket server, ta sẽ cần đến method `bind()`, chương trình server lắng nghe các port được hiện có. Đó là lý do tại sao ta cần cụ thể địa chỉ IP và PORT number.

```
server.bind(ADDR)
```

`listen()`: Server lắng nghe các yêu cầu nối kết từ các client trên cổng đã được gán.

**Giai đoạn 2:** Client tạo Socket, yêu cầu thiết lập một nối kết với Server.

- Đối với Client, không nhất thiết phải dùng `bind()` do kernel từ hệ điều hành sẽ xử lý tự động phần source IP và PORT number tạm thời.<sup>[1]</sup>

```
client.connect((SERVER_IP, PORT))
```

Thay vào đó Client dùng method `connect()`, trong đó **SERVER\_IP** và **PORT** giống như bên server để có thể kết nối được. Nếu user nhập sai **SERVER\_IP** thì không thể kết nối và phải nhập lại.

Server chấp nhận nối kết của client, khi đó một kênh giao tiếp ảo được hình thành, Client và server có thể trao đổi thông tin với nhau thông qua kênh ảo này

```
connect, address = server.accept()
```

Hàm trên trả về một cặp giá trị (connection, address)<sup>[2]</sup> mà trong đó:

- connection (trong source còn gọi là `cont`) là “socket object” mới được sử dụng để gửi và nhận data;
- address là địa chỉ gắn liền với đầu bên kia.

**Giai đoạn 3:** Trao đổi thông tin giữa Client và Server.

```
def get_msg(cont): ...  
  
def send_msg(cont, text:str):
```

- Thông thường server thực hiện hàm `get_msg` và ngghen cho đến khi có thông điệp yêu cầu (Request Message) từ client gửi đến.
- Cách nhận:

```
msg = cont.recv(int(msg_length)).decode('utf-8')
```



Method `recv()` để nhận request message từ client, đầu vào là số lượng byte cần nhận được. Sau đó mã hóa ngược các bytes này bằng `decode()`.

- Cách gửi:

```
cont.send(text.encode('utf-8'))
```

Method `send()` gửi đi các byte. Do đó ta cần mã hóa từ string sang byte bằng `encode` (mã hóa ngược cũng phải dùng chung “`utf-8`” để ra kết quả ban đầu). Chỉ có thể sử dụng với TCP dựa trên socket mà không thể với UDP

**Giai đoạn 4:** Kết thúc phiên làm việc.

Kênh ảo sẽ bị xóa khi Server hoặc Client đóng socket bằng lệnh `close()`. Nếu Client tự ý thoát mà không báo lại, server sẽ tự động xóa để sau này có thể kết nối lại.

```
cont.close()
```

**Kiểu dữ liệu của thông điệp:**

- Ở đây ta sẽ thiết kế đơn giản: một thông điệp gồm 2 phần: Header và Data. Ta sẽ gửi lần lượt từng phần này. Server sẽ nhận message Header, biết xem số byte cần đọc của Data. Dựa vào con số này sẽ đọc đúng số byte của data.
- Số byte của Header là cố định (64 bytes).
- Số byte của Data là không cố định.

**Cách tổ chức cơ sở dữ liệu:**

- Lưu account: lưu vào file txt: gồm username và password phân tách nhau bởi dấu “,”. Mỗi dòng là 1 account.
- Lưu thông tin đề tra cứu: Lưu vào file `Ty_gia_vang.json`. Ghi vào dưới dạng dictionary và đọc lên cũng vậy. Các thao tác tra cứu giống như thao tác trên dictionary.

**Các hàm xử lý:**

- Ngoại trừ 3 thư viện: `socket`, `threading`, `json` thì các thư viện còn lại sử dụng hàm nào sẽ import riêng hàm đó vào (không import tất cả).

- Các hàm khác:

<b>verify_account(username:str, password:str)</b>		
Input	Output	Mục đích: Kiểm tra trong database xem tài khoản tồn tại hay chưa
str	bool	

<b>register(username:str, password:str)</b>		
Input	Output	Mục đích: Kiểm tra tài khoản tồn tại chưa, nếu chưa thì thêm tài khoản vào database và trả về True. Ngược lại sẽ trả về False
str	bool	

*Chi tiết hơn:* Đề không yêu cầu phải dùng hàm băm mật mã, nên ta sẽ giữ nguyên.

<b>safe_split(msg, char)</b>		
Input	Output	Mục đích: cải tiến của string.split(), dùng để phân tách message từ client gửi đến. Nếu input phù hợp sẽ trả ra dạng list gồm 2 phần tử. (VD: username-password hoặc date-type). Server sẽ tiến hành xử lý trên list này.
str	list	

*Chi tiết hơn:*

<pre>print(safe_split('sinhvien', ' ')) print(safe_split('mmt hello world', ' ')) print(safe_split('mmt 20ctt3 khtn tphcm', ' '))</pre>	<pre>F:\Anaconda\python.exe [] ['mmt', 'hello world'] ['mmt', '20ctt3 khtn tphcm'] Press any key to continue . .</pre>
(input)	(output)

*Giả sử:* client nhập “**mmt hello world**” (dòng thứ 2) để đăng nhập: thì chương trình sẽ xem như “**mmt**” là username và “**hello world**” là password (Tức là username không được phép có khoảng trắng, password thì được phép)  
Đối với dòng đầu chương trình sẽ báo lỗi cho client.

<b>Các hàm trong file GUI.py</b>		
Input	Output	Mục đích: File này chứa giao diện cũng như thông báo của server. Hàm trả về str để server gửi str này đến client → Không lưu trữ giao diện tại client
	string	

<b>execution_a_client(cont, addr)</b>		
Input	Output	Mục đích: Xử lý riêng cho từng Client. Mọi hoạt động liên quan giữa server và 1 client sẽ thực hiện tại đây. Bao gồm việc xử lý đóng kết nối của 1 client
Connection		

<b>run_server()</b>		
Input	Output	Mục đích: Chạy các hàm liên quan đến server. Kèm theo việc tạo kết nối đến client mới và hủy kết nối đến tất cả các server

update_database()		
Input	Output	Mục đích: Cập nhập tỷ giá vàng. Sau mỗi lần request sẽ ghi đè lên file cũ, không lưu kiểu “append”

Chi tiết hơn:

```
with open('database/Ty_gia_vang.json', mode='w') as f:
    get_http = json.loads(get(url_links[1]).text.encode().decode('utf-8-sig'))
    f.write(json.dumps(get_http))
    f.close()
```

Ghi kiểu json và xử lý lỗi [3]:

```
json.decoder.JSONDecodeError: Unexpected UTF-8 BOM (decode using utf-8-sig): line 1 column 1 (char 0)
```

Dùng time.sleep(1800) để ngưng trong 30 phút để thực hiện việc cập nhập tiếp theo

remove_all_conts(cont, addr)		
Input	Output	Mục đích: Tuy input chỉ có 1 connection, nhưng input giống với hàm execution_a_client() nên sẽ ngắt kết nối tất cả các client khi nhập -quit() từ server
Connection		

### Nguồn tham khảo:

[1] socket.bind(): <https://pythontic.com/modules/socket/bind>

[2] socket.accept(): <https://docs.python.org/3/library/socket.html>

[3] Xử lý lỗi:

<https://www.howtosolutions.net/2019/04/python-fixing-unexpected-utf-8-bom-error-when-loading-json-data/>

Tham khảo ở tai\_lieu\_socket (python) do các giảng viên cung cấp