



# Báo cáo Socket

1. Thông tin nhóm
2. Môi trường lập trình
3. Phân công
4. Kích bản giao tiếp
5. Giải thích và Hướng dẫn sử dụng
  - Phong cách Non-blocking
  - Mô hình Event loop.
  - Phong cách blocking
  - Hướng dẫn sử dụng
6. Tài liệu tham khảo

*Đồ án có 2 phiên bản ứng dụng, trong phạm vi bản báo cáo tạm gọi là Phiên bản 1 - là phiên bản code theo phong cách non-blocking và Phiên bản 2 - phiên bản phát sinh về sau, code theo phong cách blocking*

## 1. Thông tin nhóm

Lớp 20CTT4

20120621 Nguyễn Quang Tuyến

20120524 Võ Đức Lợi

## BẢNG THỐNG KÊ HOÀN THIỆN CHỨC NĂNG

Chức năng	Non-blocking (bản 1)	Blocking (bản 2)
KẾT NỐI: - Cho phép client kết nối đến server thông qua kết nối TCP. - Cho phép client và server đặt tại các host khác nhau.	100%	100%
QUẢN LÝ KẾT NỐI: - Khi client hoặc server mất kết nối đột ngột, không làm chương trình treo hay xảy ra lỗi - Nếu một client mất kết nối không làm ảnh hưởng đến các client khác. (Quản lý kết nối đa tiểu trình ).	100%	100%
ĐĂNG NHẬP: - Client đăng nhập bằng cách gửi username, password cho server. Server nhận thông tin username, password từ client và kiểm tra với thông tin đã lưu trữ tại server.	100%	100%
ĐĂNG KÝ: - Client đăng ký bằng cách gửi username, password cho server. Server nhận thông tin username, password từ client và kiểm tra với thông tin đã lưu trữ tại server, nếu đã tồn tại, gửi thông báo đến client, yêu cầu đăng ký tài khoản khác.	100%	100%
TRA CỨU: - Cho phép Client tra cứu giá theo ngày, theo loại đồng tiền (USD, JPY, EUR ...) - Server sẽ kết nối tới một website khác (third party) để lấy thông tin (JSON hoặc HTML), sau đó rút trích thông tin và lưu trữ liệu dưới Server để phục vụ request của Client. - Server cập nhật thông tin liên tục 30 phút 1 lần của ngày hôm đó.	100%	100%
QUẢN LÝ CƠ SỞ DỮ LIỆU: - Sử dụng file cấu trúc: json	100%	100%
THOÁT: - Client được phép gửi thông báo ngừng kết nối đến server. - Server có thể gửi thông báo ngừng kết nối đến tất cả client đang hoạt động.	50% (Chưa hoàn thành chức năng: Server có thể gửi thông báo ngừng kết nối đến tất cả client đang hoạt động)	100%
GIAO DIỆN: - Có thiết kế giao diện đồ họa (GUI) cho chương trình ( Client - Server ).	100%	100%

## 2. Môi trường lập trình

Ngôn ngữ sử dụng Python. Lập trình đa nền tảng: window, linux (wsl). Thực thi được trên cả window và linux

## 3. Phân công

Nguyễn Quang Tuyến:

- Viết thư viện util gồm các module đa dụng cung cấp các API cho nhiều mục đích: loop.py, currency.py, user.py
- Viết server theo phong cách nonblocking: server\_console.py, server\_gui.py, client\_console.py
- Thiết lập mạng ảo phục vụ demo

Lưu Ngọc Quang:

- Viết client giao diện đồ họa: client\_gui.py

Võ Đức Lợi:

- Viết lại toàn bộ theo mô hình blocking. Có đủ các chức năng yêu cầu thêm một số chức năng bổ sung giúp giao diện linh hoạt hơn: Server.py và Client.py

Mức độ hoàn thành: 100%

## 4. Kịch bản giao tiếp

Server và client kết nối bằng giao thức TCP.

Cấu trúc thông điệp:

- Với phiên bản 1, thông điệp của client là các câu lệnh có syntax, thông điệp của server là kết quả trả về
- Với phiên bản 2, thông điệp được gửi theo hình thức prefixstring được chuyển từ object dictionary trong python. Prefix là độ dài của thông điệp giúp server phân loại được từng request rõ ràng, đem lại hiệu quả cao với chức năng client kiểm tra kết nối từ server.

Kiểu dữ liệu thông điệp: đều là string, đối với các request tra cứu tỉ giá, thông điệp là json-like string

Dữ liệu được lấy bằng API của 2 web:

<https://exchangeratesapi.io/>

<https://exchangerate.host/>

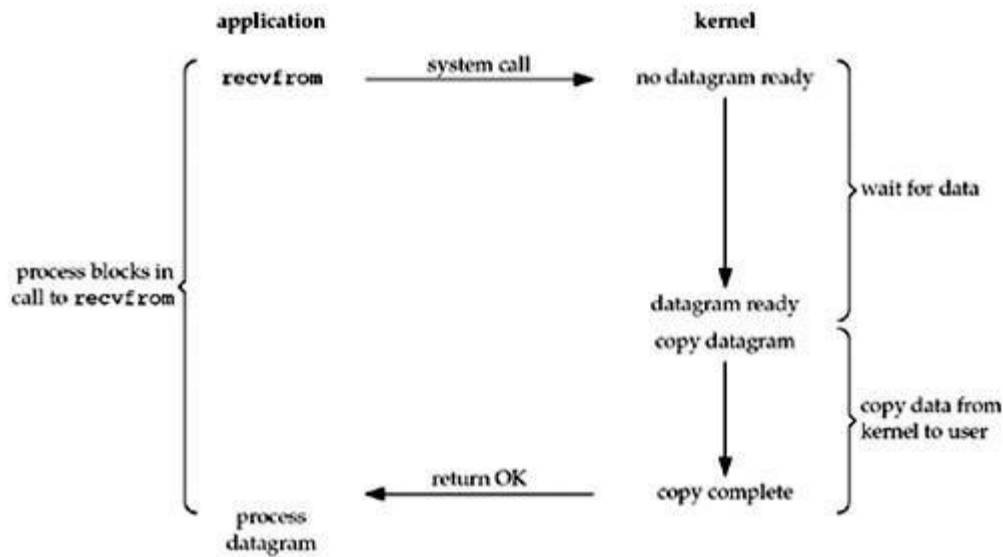
Tổ chức cơ sở dữ liệu: database được lưu theo cấu trúc:

```
{
  date1: {
    base: value,
    rates: {
      currency1: value
      currency2: value
      ...,
    }
  },
  date2: {...},
  ...
  dateN: {...}
}
```

- Với phiên bản 1, database được lưu trong 1 file json duy nhất nhằm tiện cho việc chuyển sang dict để tra cứu và cập nhật.
- Với phiên bản 2, database được lưu thành nhiều file json đại diện cho từng ngày, khi gửi cho client, data được nhóm lại thành 1 object dictionary

## 5. Giải thích và Hướng dẫn sử dụng

### Phong cách Non-blocking



Non - blocking model

Đồ án của ta có hai đặc điểm:

- Các task phải xử lý chủ yếu là input/output
- Vì chủ yếu là input/output nên thời gian xử lý task là nhỏ

Hai đặc điểm trên rất phù hợp với phong cách non-blocking. Ta chỉ cần một thread có thể linh hoạt chuyển đổi giữa các tác vụ để xử lý.

Ta cũng chỉ cần thêm một thread để làm timer đếm thời gian cập nhật database định kỳ, và một thread nữa để chạy GUI, tổng cộng là 3 thread.

Ưu điểm:

- Tận dụng tối đa hiệu suất của thread, có khả năng kết nối vô hạn

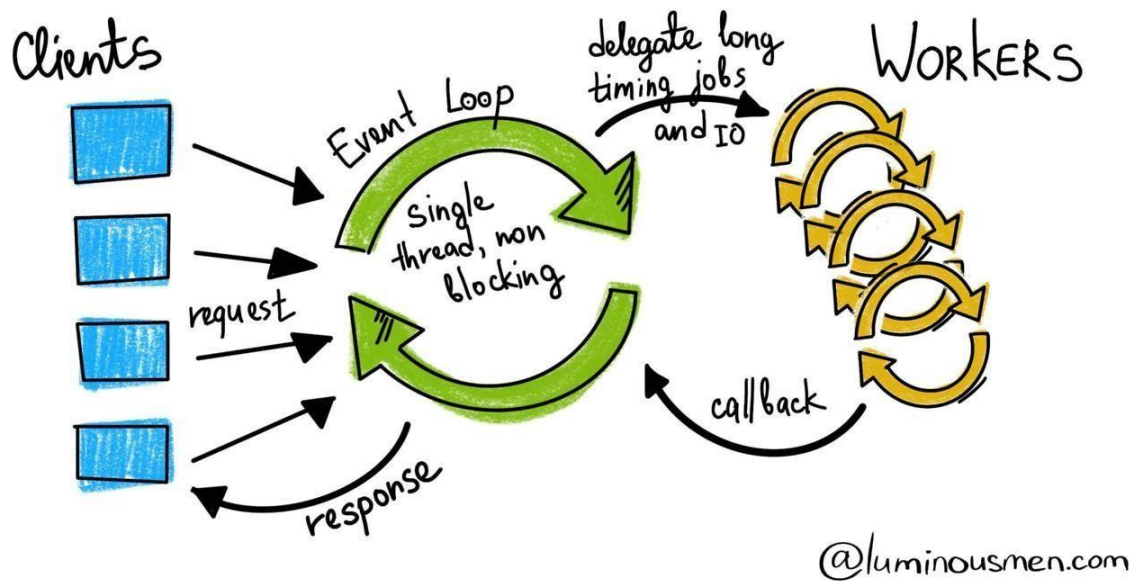
Nhược điểm:

- Khó code

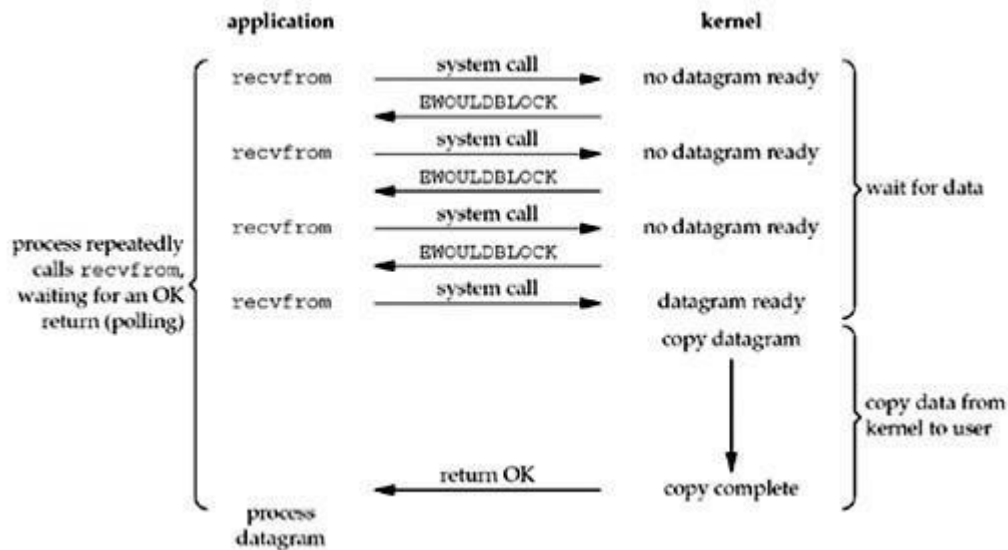
## Mô hình Event loop.

Đây là mô hình nhằm hỗ trợ cho phong cách non-blocking. Sử dụng event loop giúp code trở nên dễ đọc, dễ sử dụng và dễ quản lý hơn.

Trong đồ án này, các task trong hàm main đóng vai trò là “Workers” thực hiện các công việc input/output và class loop đóng vai trò là “Event Loop”



## Phong cách blocking



Blocking model

Vì phong cách non-blocking khó code và vẫn còn vấn đề tồn đọng, nên chúng tôi đã làm thêm một phiên bản theo phong cách blocking nhằm thỏa mãn 100% yêu cầu của giảng viên.

Đặc điểm:

- Bên phía Server, ta có 1 thread để cập nhật database sau mỗi 30 phút và ta cần thêm 1 thread khi kết nối tới Server thành công để thực hiện chờ đợi và chấp nhận kết nối từ Client. Với mỗi kết nối từ Client được chấp nhận, ta tạo thêm 1 thread để giao tiếp với Client. Bên phía Client, ta chỉ cần thêm 1 thread để kiểm tra kết nối từ Server.

Ưu điểm:

- Server này đã thỏa mãn mọi chức năng, quản lí kết nối server-client hiệu quả, tối ưu hóa thực hiện request data tới Server.

Nhược điểm:

- Với mỗi công việc input/output (với mỗi kết nối cần xử lí) bị gán chết cho một thread, do đó khả năng kết nối có giới hạn và làm hiệu suất thread bị lãng phí. Khó chạy được nhiều Client với máy có ít thread.

## Hướng dẫn sử dụng

Phiên bản 1:

Trước khi chạy file:

- Thay thế Ip address trong các file thành Ip address phù hợp
- Trong file `server_gui.py` thay thế `"from util.loop import Loop"` thành `"from util.loop_window import Loop"` nếu chạy trên window.

Giao diện client có các tính năng tra cứu ở mỗi layer, được phân loại dựa theo các API của bên thứ 3, gồm:

- latest: tra cứu tỉ giá mới nhất và cập nhật database
- historical: tra cứu tỉ giá theo ngày (cập nhật nếu database chưa có)
- convert: chuyển đổi tỉ giá tiền tệ (theo ngày) (cập nhật nếu database chưa có)
- timeseries: tra cứu tỉ giá trong một khoảng thời gian (cập nhật nếu database chưa có)

Ngoài các chức năng chính còn một số chức năng phải có như đăng nhập đăng ký, đăng xuất, nút quit.

Phiên bản 2:

Trước khi chạy file:

- Thay thế Ip address và port trong các file thành Ip address và port phù hợp, mặc định trong file.py và file.exe là “127.0.0.1” và 5566.
- Thay thế ngày bắt đầu lấy data từ API trong file Server.py (biến DATE) bởi vì API cung cấp tỉ giá tiền tệ tới năm 1999 nhưng nếu lấy hết thì database nặng và vượt quá số request API cho phép (1000 request – 1 request / 1 ngày). Mặc định trong file.py và file.exe là DATE = “2021-01-01”, tức là tra cứu tỉ giá tiền tệ cho tới 01/01/2021.

Giao diện Server có các tính năng sau

- Checkbox ‘Automatic’: Tự động điền IpAddress và Port mà đã chọn từ đầu làm địa chỉ liên kết socket.
- Button ‘Connect’: Kết nối tới địa chỉ được chọn trước để liên kết với socket và đợi kết nối từ các client.
- Button ‘Stop’: Tạm dừng kết nối tới Server và thông báo ngắt kết nối tới các client
- Button ‘Clear’: Xóa hết các thông báo trong hộp thông báo
- Hộp thông báo: Hiện thị các thông báo của server và clients
- Hộp Status: Hiện thị các client đang kết nối và username được đăng nhập bởi client
- Các Client không ảnh hưởng nhau và nếu bất kì Client nào ngắt kết nối tới Server hoặc bị tắt app đột ngột, Server sẽ nhận được thông báo và cập nhật lại hộp status.

Giao diện Client có các tính năng sau

- Ở trang kết nối, phải nhập IpAddress và Port để kết nối tới Server, nếu nhập sai hoặc Server chưa được kết nối sẽ hiện thị trang 404. Kết nối thành công chuyển qua trang đăng nhập
- Checkbox ‘Automatic’: Tự động điền IpAddress và Port mà đã chọn từ đầu để kết nối Server
- Ở trang đăng nhập, có thể đăng nhập hoặc đăng kí tài khoản, sẽ hiện thị thông báo nếu tài khoản và mật khẩu không hợp lệ. Đăng nhập thành công chuyển sang trang tra cứu
- Checkbox “Show password”: Dùng để ẩn/hiển thị mật khẩu nhập vào
- Button ‘DISCONNECT’: Tắt kết nối tới Server, gửi thông báo cho Server và quay lại trang kết nối.
- Ở trang tra cứu, khi vừa đăng nhập thành công sẽ hiện lên tất cả data tỉ giá tiền tệ từ ngày bắt đầu lấy data (mặc định là 2021-01-01), thực hiện tra cứu



tỉ giá tiền tệ theo 5 trường: năm, tháng, ngày, base (tiền tệ cơ sở), currency (tiền tệ muốn tra).

- Button 'DISCONNECT': Tắt kết nối tới Server, gửi thông báo cho Server và quay lại trang kết nối.
- Button 'LOGOUT': Đăng xuất khỏi tài khoản hiện tại, gửi thông báo cho Server và quay về trang đăng nhập.
- Button 'Search': Thực hiện tra cứu theo 5 trường đã chọn, nếu ngày tra không hợp lệ hoặc ngày tra vượt qua ngày bắt đầu lấy data (biến DATE) thì sẽ hiện ra thông báo.
- Button 'Reset': Lấy data mới nhất từ Server và hiện thị tất cả lên màn hình như lúc mới đăng nhập thành công, đặt lại giá trị các trường phù hợp.
- Góc dưới bên trái của trang có ghi chú data được lấy bắt đầu từ ngày nào để thuận tiện tra cứu.
- Nếu Server ngắt kết nối hoặc tắt app đột ngột, tất cả Client sẽ nhận được thông báo và quay trở về trang kết nối ban đầu. Không ảnh hưởng app đang chạy.

## 6. Tài liệu tham khảo

**Blocking and Non-blocking I/O**

**I/O models**

**A Hitchhikers Guide to Asynchronous Programming**

**PYTHON - GUI Programming (Tkinter)**

**NAT, Bridged and Internal Networking in VirtualBox**

**More details about virtualbox networking settings**