

# ĐỒ ÁN: LẬP TRÌNH SOCKET REPORT

Tài liệu này mô tả nội dung đồ án môn học cho môn học Mạng máy tính

Nhóm 9

22127230 – Dương Bình Nguyễn Lân

22127420 – Nguyễn Hà Nam Trân

22127476 – Đặng Triệu Kha

22127486 – Đào Ngọc Thiện



Khoa Công nghệ Thông tin  
Đại học Khoa học Tự nhiên TP HCM  
Tháng Thg8-23

# MỤC LỤC

<b>1</b>	<b>Tổng quan.....</b>	<b>1</b>
	Thông tin nhóm.....	1
	Thông tin đề án.....	1
<b>2</b>	<b>Nội dung đề án .....</b>	<b>2</b>
2.1	File Organization.....	2
2.2	Class and function.....	3
	Server: server.py.....	3
	Client: client.py.....	5
	List application: listApp.py.....	7
	Start application: startApp.py.....	8
	Start process: start.py.....	9
	Kill process/Kill application: kill.py.....	10
	List processs: process.py.....	11
	Key stroke: keylog.py.....	12
	Screen capture: pic.py.....	13
2.3	Hướng dẫn sử dụng sản phẩm.....	13
<b>3</b>	<b>Đánh giá.....</b>	<b>14</b>
3.1	Đánh giá mức độ hoàn thành.....	14
3.2	Đánh giá tỷ lệ đóng góp.....	14

# 1

## Tổng quan

### Thông tin nhóm

MSSV	Họ tên	Email	Vai trò
22127230	Dương Bình Nguyễn Lân	dbnlan22@clc.fitus.edu.vn	Developer
22127420	Nguyễn Hà Nam Trân	nhntran22@clc.fitus.edu.vn	Developer
22127476	Đặng Triệu Kha	dtkha22@clc.fitus.edu.vn	Developer
22127486	Đào Ngọc Thiện	dnthien22@clc.fitus.edu.vn	Developer

### Thông tin đồ án

Tên đồ án	
Công cụ hỗ trợ	ChatGPT
	Microsoft Word
	Canva
	Stackoverflow
Product Owner	Đỗ Hoàng Cường

# 2

## Nội dung đồ án

### 2.1 File Organization

Các file được tổ chức thành 2 folder là Server và Client:

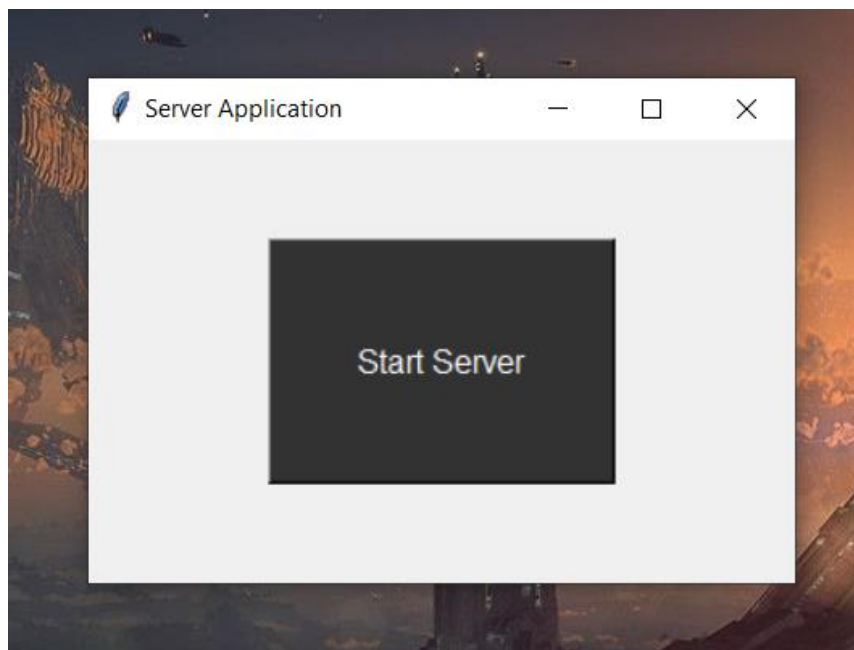
- **Server:** Folder này chứa:
  - Các file temp (do hệ thống tự tạo) từ các lần chạy.
  - Các file ảnh chụp màn hình được tạo ra từ mỗi lần chạy lệnh chụp màn hình.
  - File **keylogger.txt** chứa dữ liệu bắt phím nhấn. Nếu file này không có thì sau mỗi lần chạy sẽ được tự động tạo.
  - File **server.py**: Chứa các hàm dùng để chờ nhận kết nối từ client và thiết lập kết nối với client. Ngoài ra còn xử lý và trả lời các yêu cầu được gửi đi từ client.
- **Client:** Folder này chứa:
  - Các file temp (do hệ thống tự tạo) từ các lần chạy.
  - Các file ảnh chụp màn hình được tạo ra từ mỗi lần chạy lệnh chụp màn hình.
  - 8 file xử lý chính:
    - **client.py**: Chứa các hàm để gửi, yêu cầu các lệnh cho server.
    - **keylog.py**: Chứa các hàm để xử lý cho việc bắt phím nhấn.
    - **kill.py**: Chứa các hàm để tắt tiến trình hoặc ứng dụng.
    - **listApp.py**: Chứa các hàm để liệt kê, bật, tắt các ứng dụng có trong máy server.
    - **process.py**: Chứa các hàm để liệt kê, bật, tắt các tiến trình có trong máy server.
    - **pic.py**: Chứa các hàm để xử lý việc lấy ảnh chụp màn hình từ server và hiển thị lên màn hình client.
    - **start.py**: Chứa các hàm để bật tiến trình.
    - **startApp.py**: Chứa các hàm để bật ứng dụng.

## 2.2 Class and function

Hầu hết các hàm trong tất cả các class của các file, có một tham số truyền vào được đặt tên là "self". Tham số này ám chỉ chính đối tượng class chứa hàm có tham số self đó, tham số "self" được dùng để tạo ra các đối tượng có thể được sử dụng ở các hàm khác. Ngoài ra còn một tham số khác là "root" (nhưng ít xuất hiện hơn "self"), tham số này ám chỉ giao diện hiện tại (Ví dụ: Người dùng đang ở file "client.py" thì tham số "root" ám chỉ giao diện chính ở bên phía Client, người dùng đang ở file "process.py" thì tham số "root" ám chỉ giao diện tiến trình). Tham số "root" hỗ trợ thiết kế giao diện, tạo các đối tượng trên giao diện như nút bấm hay ô lấy dữ liệu từ người dùng.

Trong một số file bên phía client, ở hàm \_\_init\_\_ có truyền vào 2 tham số là "client" và "client\_socket". Hai tham số này lần lượt tương ứng với tham số "self" và "client\_socket" trong file "client.py". Nghĩa là hai tham số này đóng vai trò như là các biến để truyền kết nối client – server tới file hiện tại, nếu không có 2 tham số này, dữ liệu từ các file không phải "client.py" và "server.py" sẽ không thể gửi và nhận.

### 2.2.1 Server: server.py

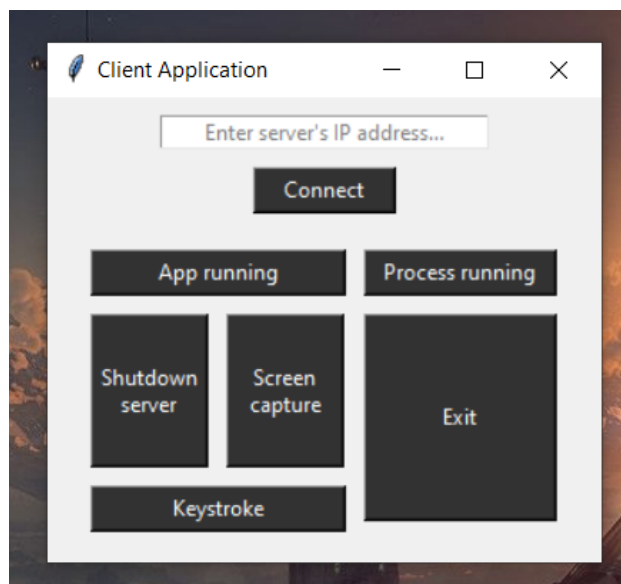


Class **ServerApp**:

- **\_\_init\_\_(self, root)**: Khởi tạo giao diện của Server Application, giao diện này có 1 nút bấm tương ứng với chức năng kết nối với client. Trong hàm này có 2 hàm phụ:
  - **center\_widget(widget, root)**: Tự động điều chỉnh kích thước của nút bấm khi thay đổi kích thước giao diện của server. Tham số "widget" ở đây ám chỉ nút bấm đang cần được tự động thay đổi kích thước.
  - **on\_window\_resize(event)**: Dùng để cố định với hành động thay đổi kích thước của giao diện, nghĩa là mỗi khi người dùng thay đổi kích thước của giao diện thì hàm này sẽ được gọi. Tham số "event" chỉ sự kiện thay đổi kích thước <Configure>.
- **start\_server(self)**: Được gọi khi người dùng nhấn vào nút Start Server, chờ đợi kết nối từ client và thiết lập kết nối với client, gọi các hàm tương ứng với các tín hiệu mà client đã gửi qua.
- **takepic(self)**: Thực hiện thao tác chụp màn hình và gửi dữ liệu ảnh vừa chụp tới client.
- **clear\_store\_file(self)**: Xóa dữ liệu của file keylogger.txt.
- **save\_key\_event(self, e)**: Khi có sự kiện ở bàn phím xảy ra, hàm save\_key\_event sẽ được gọi, nó sẽ lưu phím vừa được nhấn vào file keylogger.txt. Tham số "e" ở đây là sự kiện phím nhấn vừa xảy ra.
- **hook\_key(self)**: Bắt đầu theo dõi các thao tác nhấn phím từ server với hàm save\_key\_event.
- **unhook(self)**: Ngừng theo dõi các thao tác phím nhấn.
- **send\_data(self, data)**: Gửi kích cỡ dữ liệu và dữ liệu. Tham số "data" là dữ liệu được gửi đi.
- **printKeylogger(self)**: Lấy dữ liệu từ file keylogger.txt và chuyển dữ liệu theo dõi được tới máy client bằng hàm send\_data().
- **keylog(self)**: Nhận tín hiệu từ client và gọi các hàm thực hiện theo dõi phím nhấn tương ứng với các tín hiệu đó.
- **list\_apps(self)**: Liệt kê danh sách các ứng dụng trong máy server rồi gửi về cho client.

- **kill\_apps(self, process\_id)**: Tắt ứng dụng được chỉ định ở trên server và gửi thông tin về việc liệu ứng dụng đó đã được tắt thành công hay bị lỗi cho client. Tham số "process\_id" ám chỉ mã định danh của ứng dụng cần được tắt.
- **start\_program(self, appName)**: Bật ứng dụng được chỉ định ở trên server và gửi thông tin về việc bật thành công hay thất bại cho client. Tham số "appName" chỉ tên ứng dụng cần được bật.
- **application(self)**: Nhận dữ liệu từ client để thực hiện các hành động bật, tắt, xem các ứng dụng trên máy server.
- **list\_process(self)**: Liệt kê danh sách các tiến trình trong máy server, gửi về cho client.
- **kill\_process(self, process\_id)**: Tắt các tiến trình được chỉ định ở trên server và gửi thông tin về việc liệu tiến trình đó được tắt thành công hay bị lỗi cho client. Tham số "process\_id" ám chỉ mã định danh của tiến trình đang cần được tắt.
- **get\_exe\_files(self, folder\_path)**: Trả về danh sách các file có đuôi .exe trong máy server.
- **start\_program(self, dir\_program\_name)**: Bật tiến trình dựa vào đường dẫn tới file .exe của tiến trình đó. Tham số "dir\_program\_name" chỉ đường dẫn tới tiến trình cần được tắt.
- **process(self)**: Nhận dữ liệu từ client để thực hiện các hành động bật, tắt, xem các tiến trình trên máy server.
- **shutdown(self)**: Gọi lệnh tắt máy bên phía server.

## Client: client.py



Class **ClientApp**:

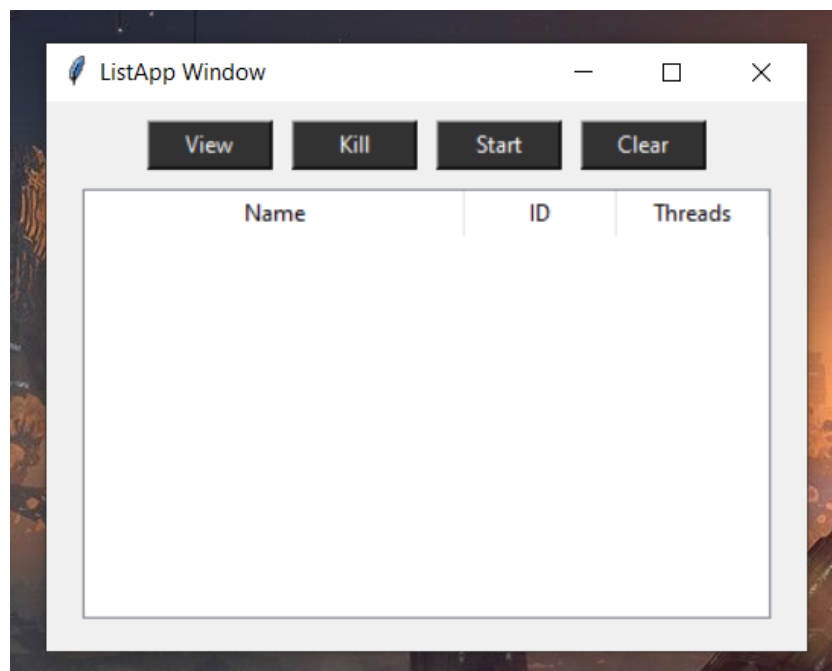
- **\_\_init\_\_(self, root)**: Khởi tạo giao diện Client Application, giao diện này bao gồm 1 ô input và 7 nút bấm, tương ứng với 7 chức năng: kết nối, thao tác với ứng dụng, thao tác với tiến trình, tắt máy, chụp màn hình, thực hiện bắt phím nhấn, và đóng kết nối. Trong hàm này có 2 hàm phụ:
  - **on\_entry\_click(event)**: Hàm này được gọi lên khi người dùng ấn chuột vào ô nhập địa chỉ IP, dùng để xóa dòng chữ mặc định: "Enter server's IP address..." và đổi màu chữ thành màu đen. Tham số "event" ám chỉ sự kiện click vào ô input.
  - **on\_entry\_focus\_out(event)**: Hàm này được gọi lên khi người dùng ấn chuột vào bất kì vị trí nào nằm ngoài ô input, dùng để hiện lại dòng chữ mặc định: "Enter server's IP address..." và đổi màu chữ thành màu xám. Tham số "event" ám chỉ sự kiện click vào vị trí ngoài ô input.
- **connect\_to\_server(self)**: Lấy dữ liệu IP mà người dùng đã nhập từ ô input và gửi kết nối tới thiết bị có địa chỉ IP đó, sau một khoảng thời gian sẽ thông báo liệu kết nối đã được thiết lập thành công hay chưa. Hàm này được gọi khi người dùng ấn vào nút "Connect".
- **open\_application(self)**: Hàm này được gọi khi người dùng ấn vào nút "App running". Nếu kết nối chưa được thiết lập hoặc thiết lập không thành công, hàm sẽ gửi tín hiệu cảnh báo kết nối chưa được thiết lập. Mặt khác, nếu kết nối đã được thiết lập, hàm sẽ khởi tạo một giao diện "listApp" từ file "listApp.py" và đưa người dùng tới giao diện đó.
- **shutdown\_server(self)**: Hàm này được gọi khi người dùng ấn vào nút "Shutdown server". Nếu kết nối chưa được thiết lập hoặc thiết lập không thành công, hàm sẽ gửi tín hiệu cảnh báo kết nối chưa được thiết lập. Mặt khác, nếu kết nối đã được thiết lập, hàm sẽ gửi tín hiệu yêu cầu tắt máy tới server, và tự đóng kết nối. Sau đó sẽ hiển thị thông báo server đã được tắt.
- **close\_connection(self)**: Hàm này được gọi khi người dùng ấn vào nút "Exit". Nếu kết nối chưa được thiết lập hoặc thiết lập không thành công, hàm sẽ gửi tín hiệu cảnh báo kết nối chưa được thiết lập. Mặt khác, nếu kết nối đã được thiết lập,



hàm sẽ gửi tín hiệu đóng kết nối tới server. Sau đó sẽ tự đóng kết nối và tắt giao diện client.

- **take\_picture(self):** Hàm này được gọi khi người dùng ấn vào nút "Screen capture". Nếu kết nối chưa được thiết lập hoặc thiết lập không thành công, hàm sẽ gửi tín hiệu cảnh báo kết nối chưa được thiết lập. Mặt khác, nếu kết nối đã được thiết lập, hàm sẽ gọi đối tượng "screenCaptureWindow" từ file "pic.py", sau đó gửi tín hiệu yêu cầu chụp ảnh màn hình và hiển thị ảnh màn hình vừa chụp.
- **start\_keylogging(self):** Hàm này được gọi khi người dùng ấn vào nút "Keystroke". Nếu kết nối chưa được thiết lập hoặc thiết lập không thành công, hàm sẽ gửi tín hiệu cảnh báo kết nối chưa được thiết lập. Mặt khác, nếu kết nối đã được thiết lập, hàm sẽ khởi tạo một giao diện "keylogWindow" từ file "keylog.py" và đưa người dùng tới giao diện đó.
- **list\_processes(self):** Hàm này được gọi khi người dùng ấn vào nút "Process running". Nếu kết nối chưa được thiết lập hoặc thiết lập không thành công, hàm sẽ gửi tín hiệu cảnh báo kết nối chưa được thiết lập. Mặt khác, nếu kết nối đã được thiết lập, hàm sẽ khởi tạo một giao diện "listProcess" từ file "process.py" và đưa người dùng tới giao diện đó.

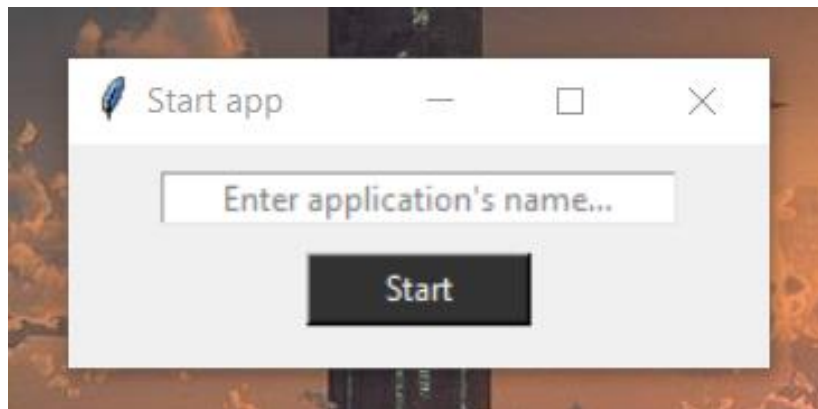
## List application: listApp.py



Class **listApp**:

- **\_\_init\_\_(self, client, client\_socket)**: Khởi tạo giao diện listApp. Giao diện này bao gồm 1 hộp chứa thông tin ứng dụng và 4 nút bấm tương ứng với các chức năng: xem, bật, tắt ứng dụng và xóa danh sách các ứng dụng trong hộp thông tin.
- **view\_apps(self)**: Gửi tín hiệu yêu cầu danh sách các ứng dụng đang chạy cho server và nhận dữ liệu về danh sách các ứng dụng đó từ phía server. Sau đó hiển thị danh sách các ứng dụng vừa nhận lên hộp thông tin. Hàm này được gọi khi người dùng ấn vào nút "View".
- **kill\_app(self)**: Gửi tín hiệu tắt ứng dụng và khởi tạo đối tượng "KillProcess" từ file "kill.py". Hàm này được gọi khi người dùng ấn vào nút "Kill".
- **start\_app(self)**: Gửi tín hiệu bật ứng dụng và khởi tạo đối tượng "StartApp" từ file "startApp.py". Hàm này được gọi khi người dùng ấn vào nút "Start".
- **clear\_list(self)**: Xóa danh sách các ứng dụng đang hiển thị trên hộp thông tin. Hàm này được gọi khi người dùng ấn vào nút "Clear".
- **on\_closing(self)**: Gửi tín hiệu thông báo cho server thoát khỏi chức năng "List app" và tắt giao diện "listApp" bên phía client.

### Start application: startApp.py

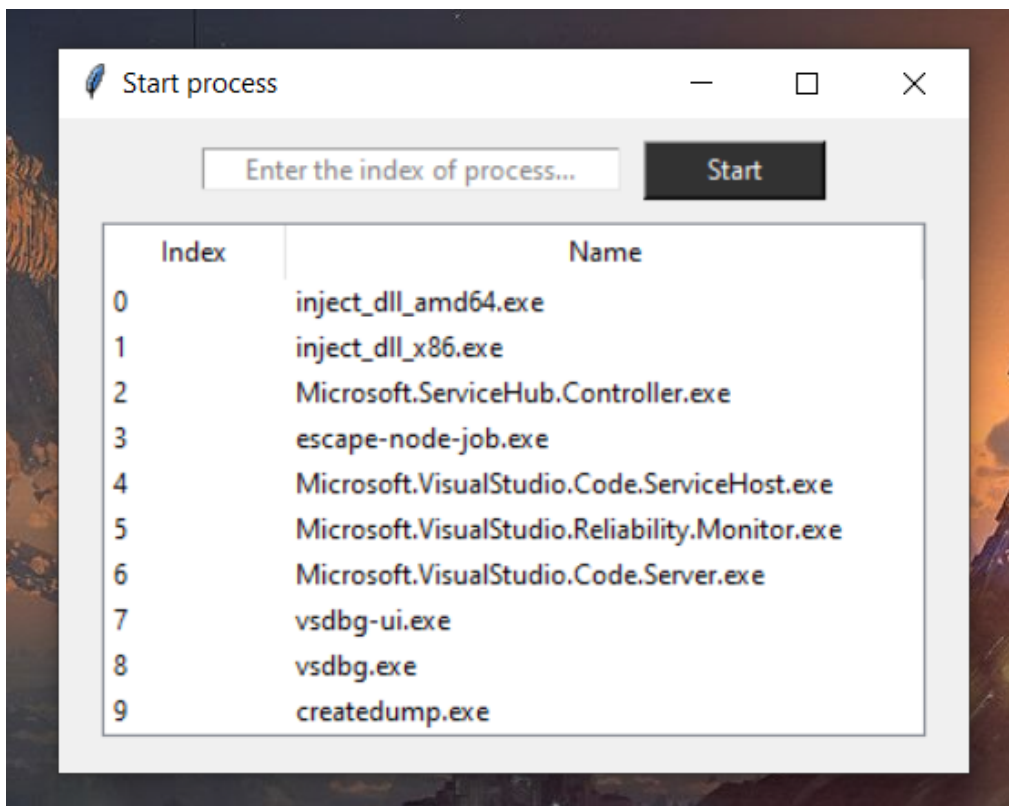


Class **StartApp**:

- **\_\_init\_\_(self, client, client\_socket)**: Khởi tạo giao diện Start app. Giao diện này bao gồm 1 ô input và 1 nút bấm tương ứng với chức năng bật ứng dụng. Trong hàm này có 2 hàm phụ:

- **on\_entry\_click(event)**: Hàm này được gọi lên khi người dùng ấn chuột vào ô nhập địa chỉ IP, dùng để xóa dòng chữ mặc định: "Enter application's name..." và đổi màu chữ thành màu đen. Tham số "event" ám chỉ sự kiện click vào ô input.
- **on\_entry\_focus\_out(event)**: Hàm này được gọi lên khi người dùng ấn chuột vào bất kì vị trí nào nằm ngoài ô input, dùng để hiện lại dòng chữ mặc định: "Enter application's name..." và đổi màu chữ thành màu xám. Tham số "event" ám chỉ sự kiện click vào vị trí ngoài ô input.
- **send\_start\_request(self)**: Lấy tên ứng dụng mà người dùng đã nhập trong ô input và gửi lệnh yêu cầu khởi chạy ứng dụng đó cho server. Sau đó nhận tín hiệu từ phía server xem ứng dụng đã được khởi chạy thành công hay chưa. Hàm này được gọi khi người dùng ấn vào nút "Start".
- **on\_closing(self)**: Gửi tín hiệu thông báo cho server thoát khỏi chức năng "Start app" và tắt giao diện "Start app" bên phía client.

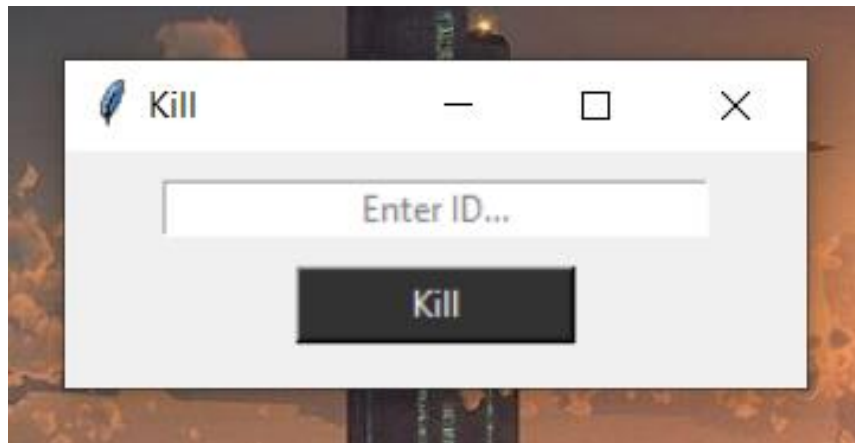
## Start process: start.py



### Class **StartProcess**:

- **\_\_init\_\_(self, client, client\_socket)**: Khởi tạo giao diện Start process. Giao diện này bao gồm 1 hộp chứa danh sách các tiến trình được gợi ý, 1 ô input và 1 nút bấm tương ứng với chức năng bật tiến trình. Do tên tiến trình khá dài và một số tên rất khó để nhớ nên hộp thông tin này được tạo ra nhằm gợi ý cho người dùng một danh sách các tiến trình trong máy. Người dùng chỉ cần nhập chỉ số tương ứng với tiến trình muốn bật và ấn nút "Start".
- **send\_start\_request(self)**: Hàm này được gọi lên khi người dùng ấn vào nút "Start". Hàm sẽ lấy chỉ số của tiến trình mà người dùng đã nhập trong ô input và gửi lệnh yêu cầu khởi chạy tiến trình tương ứng với chỉ số đó cho server. Sau đó nhận tín hiệu từ phía server xem tiến trình đã được khởi chạy thành công hay chưa.
- **on\_closing(self)**: Gửi tín hiệu thông báo cho server thoát khỏi chức năng "Start process" và tắt giao diện "Start process" bên phía client.

### Kill process/Kill application: kill.py

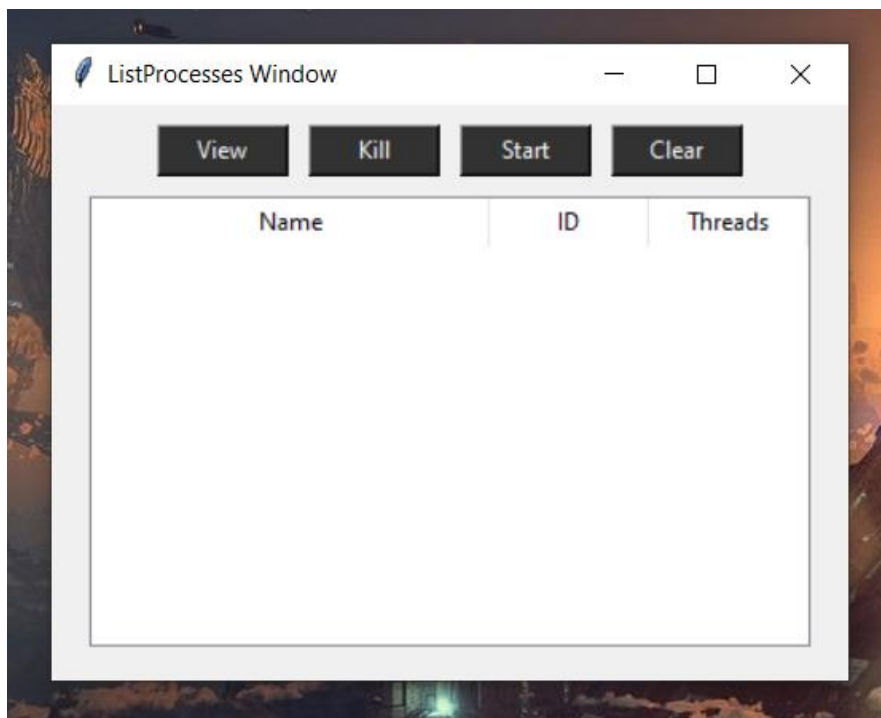


### Class **KillProcess**:

- **\_\_init\_\_(self, client, client\_socket)**: Khởi tạo giao diện Kill. Giao diện này bao gồm 1 ô input và 1 nút bấm tương ứng với chức năng tắt tiến trình hoặc ứng dụng. Trong hàm này có 2 hàm phụ:
  - **on\_entry\_click(event)**: Hàm này được gọi lên khi người dùng ấn chuột vào ô nhập địa chỉ IP, dùng để xóa dòng chữ mặc định: "Enter ID..." và đổi màu chữ thành màu đen. Tham số "event" ám chỉ sự kiện click vào ô input.

- **on\_entry\_focus\_out(event)**: Hàm này được gọi lên khi người dùng ấn chuột vào bất kì vị trí nào nằm ngoài ô input, dùng để hiện lại dòng chữ mặc định: "Enter ID..." và đổi màu chữ thành màu xám. Tham số "event" ám chỉ sự kiện click vào vị trí ngoài ô input.
- **send\_kill\_request(self, client)**: Hàm này được gọi lên khi người dùng ấn vào nút "Kill". Hàm sẽ lấy chỉ số của tiến trình/ứng dụng mà người dùng đã nhập trong ô input và gửi lệnh yêu cầu tắt tiến trình/ứng dụng tương ứng với chỉ số đó cho server. Sau đó nhận tín hiệu từ phía server xem tiến trình/ứng dụng đã được tắt thành công hay chưa.
- **on\_closing(self, client)**: Gửi tín hiệu thông báo cho server thoát khỏi chức năng "Kill" và tắt giao diện "Kill" bên phía client.

## List processs: process.py

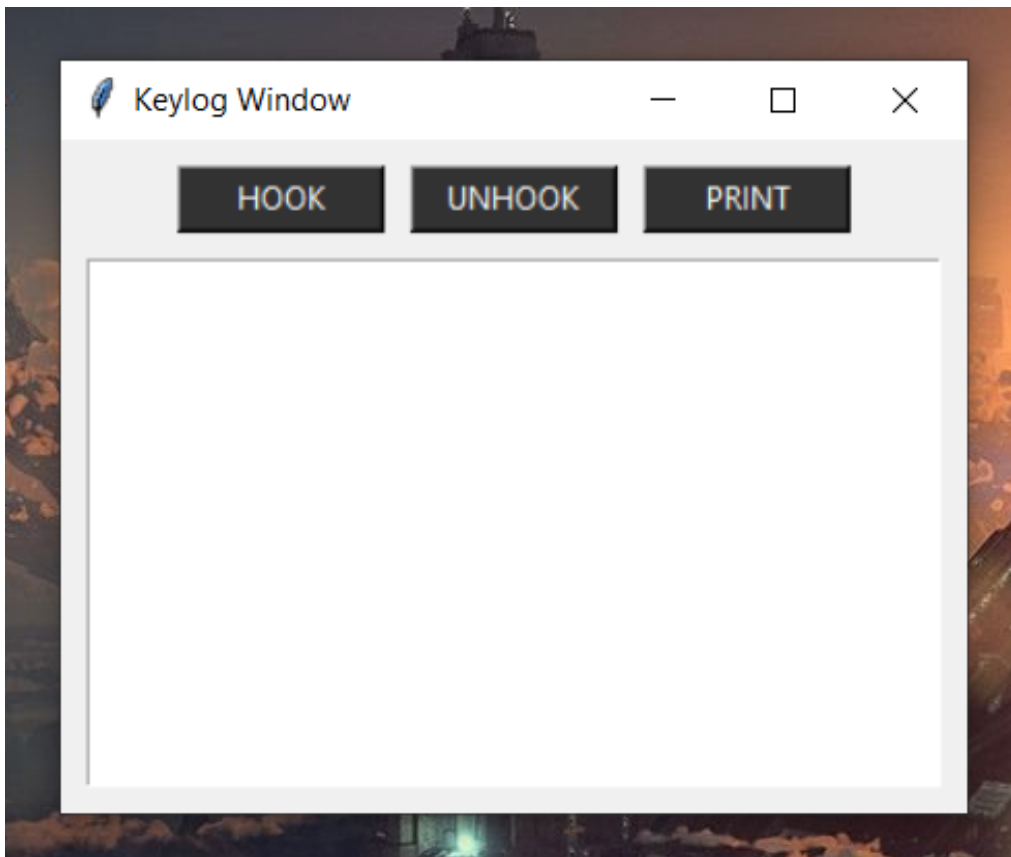


### Class **listProcess**:

- **\_\_init\_\_(self, client, client\_socket)**: Khởi tạo giao diện listProcesses. Giao diện này bao gồm 1 hộp chứa thông tin tiến trình và 4 nút bấm tương ứng với các chức năng: xem, bật, tắt tiến trình và xóa danh sách các tiến trình trong hộp thông tin.

- **view\_processes(self)**: Gửi tín hiệu yêu cầu danh sách các tiến trình đang chạy cho server và nhận dữ liệu về danh sách các tiến trình đó từ phía server. Sau đó hiển thị danh sách các tiến trình vừa nhận lên hộp thông tin. Hàm này được gọi khi người dùng ấn vào nút "View".
- **kill\_processes(self)**: Gửi tín hiệu tắt tiến trình và khởi tạo đối tượng "KillProcess" từ file "kill.py". Hàm này được gọi khi người dùng ấn vào nút "Kill".
- **start\_processes(self)**: Gửi tín hiệu bật tiến trình và khởi tạo đối tượng "StartProcess" từ file "start.py". Hàm này được gọi khi người dùng ấn vào nút "Start".
- **clear\_processes(self)**: Xóa danh sách các ứng dụng đang hiển thị trên hộp thông tin. Hàm này được gọi khi người dùng ấn vào nút "Clear".
- **on\_closing(self)**: Gửi tín hiệu thông báo cho server thoát khỏi chức năng "List process" và tắt giao diện "listProcesses" bên phía client.

### Key stroke: keylog.py



### Class **KeylogWindow**:

- **\_\_init\_\_(self, client, client\_socket)**: Khởi tạo giao diện Keylog Window. Giao diện này bao gồm 1 hộp chứa thông tin tiến trình và 3 nút bấm tương ứng với các chức năng: bắt đầu hành động bắt phím nhấn, dừng hành động bắt phím nhấn và hiển thị danh sách các phím nhấn vừa bắt được trên hộp thông tin.
- **send\_hook(self)**: Gửi tín hiệu bắt đầu hành động bắt phím nhấn cho server. Hàm này được gọi khi người dùng ấn vào nút "HOOK".
- **send\_unhook(self)**: Gửi tín hiệu dừng hành động bắt phím nhấn cho server. Hàm này được gọi khi người dùng ấn vào nút "UNHOOK".
- **recvall(self)**: Nhận một chuỗi dữ liệu tương ứng với chiều dài của chuỗi dữ liệu được mã hóa ở bên server. Sau đó nhận đúng số dữ liệu được mã hóa mà server gửi qua và trả về dữ liệu được giải mã.
- **send\_print(self)**: Gửi tín hiệu yêu cầu truyền chuỗi dữ liệu vừa theo dõi được cho server. Sau đó hiển thị chuỗi dữ liệu vừa nhận trên hộp thông tin. Hàm này được gọi khi người dùng ấn vào nút "PRINT".
- **on\_closing(self)**: Gửi tín hiệu thông báo cho server thoát khỏi chức năng "Keylog" và tắt giao diện "Keylog Window" bên phía client.

### Screen capture: **pic.py**

#### Class **screenCaptureWindow**:

- **\_\_init\_\_(self, client, client\_socket)**: Khởi tạo các biến client và client.client\_socket của đối tượng screenCaptureWindow.
- **request\_screenshot(self)**: Gửi tín hiệu yêu cầu chụp màn hình cho server, sau đó nhận dữ liệu ảnh được gửi từ server. Lưu các dữ liệu đó thành ảnh dưới dạng file .png và hiển thị lên màn hình.
- **on\_closing(self)**: Gửi tín hiệu thông báo cho server thoát khỏi chức năng "Screen capture".

## 2.3 Hướng dẫn sử dụng sản phẩm

Link video hướng dẫn sử dụng sản phẩm:

<https://www.youtube.com/watch?v=IGUwFYz79Nw>

# 3

## Đánh giá

### 3.1 Đánh giá mức độ hoàn thành

Công việc	Mức độ hoàn thành
Screen capture	100%
Key logger	100%
List app	100%
List process	100%
Shutdown	100%

### 3.2

### Đánh giá tỷ lệ đóng góp

MSSV	Họ tên	Tỷ lệ	Ghi chú
22127230	Dương Bình Nguyễn Lâm	100%	
22127420	Nguyễn Hà Nam Trân	100%	
22127476	Đặng Triệu Kha	100%	
22127486	Đào Ngọc Thiện	100%	