**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

---o0o---

Logo

Description automatically generated with medium confidence

**BÁO CÁO**

**ĐỒ ÁN LẬP TRÌNH SOCKET:**

**MÔ HÌNH CLIENT - SERVER TRÊN NGÔN NGỮ LẬP TRÌNH PYTHON**

**Sinh viên MSSV**

Nguyễn Ngọc Khôi Nguyên 19120106

Nguyễn Võ Anh Khoa 19120256

**GVHD:** Thầy Đỗ Hoàng Cường

**Lớp:** 19CNTN

**Môn học:** Mạng máy tính

Ngày 6 tháng 11 năm 2021

# Mục lục

Contents

[I. Mục lục 2](#_Toc87232628)

[II. Môi trường làm việc, ngôn ngữ, thư viện hỗ trợ 4](#_Toc87232629)

[1. Môi trường làm việc: 4](#_Toc87232630)

[2. Ngôn ngữ: 4](#_Toc87232631)

[3. Thư viện hỗ trợ: 4](#_Toc87232632)

[III. Mô tả các file trong source code 4](#_Toc87232633)

[1. Thư mục Client 4](#_Toc87232634)

[2. Thư mục Server 4](#_Toc87232635)

[IV. Hướng dẫn sử dụng và tính năng 5](#_Toc87232636)

[1. Kết nối Client-Server 5](#_Toc87232637)

[2. Tab APPS CONTROLLER 6](#_Toc87232638)

[a. Refresh 6](#_Toc87232639)

[b. Start 7](#_Toc87232640)

[c. Kill 7](#_Toc87232641)

[3. Tab PROCESSES CONTROLLER 8](#_Toc87232642)

[a. Refresh 9](#_Toc87232643)

[b. Start 9](#_Toc87232644)

[c. Kill 10](#_Toc87232645)

[4. Tab FTP CONTROLLER 10](#_Toc87232646)

[a. Hiển thị đường dẫn 11](#_Toc87232647)

[b. Delete Folder(File) trên Client 11](#_Toc87232648)

[c. Send Folder(File) từ Client tới Server 12](#_Toc87232649)

[d. Delete Folder(File) trên Server 12](#_Toc87232650)

[e. Lấy Folder(File) từ Server sang Client 13](#_Toc87232651)

[5. Tab KEYBOARD CONTROLLER 13](#_Toc87232652)

[a. Lock 13](#_Toc87232653)

[b. Hook 14](#_Toc87232654)

[c. Unhook 14](#_Toc87232655)

[d. Print 14](#_Toc87232656)

[e. Delete 14](#_Toc87232657)

[6. Tab MAC CONTROLLER 14](#_Toc87232658)

[7. Tab POWER CONTROLLER 15](#_Toc87232659)

[a. Log out 16](#_Toc87232660)

[b. Shutdown 16](#_Toc87232661)

[8. Tab STREAMING CONTROLLER 16](#_Toc87232662)

[9. Tab REGISTRY CONTROLLER 17](#_Toc87232663)

[a. Sửa bằng File Registry 17](#_Toc87232664)

[b. Sửa trực tiếp 18](#_Toc87232665)

[10. Disconnect 18](#_Toc87232666)

[V. Mô tả mã nguồn đồ án 19](#_Toc87232667)

[2. Connect 19](#_Toc87232668)

[a. Client 19](#_Toc87232669)

[b. Server 19](#_Toc87232670)

[3. Message chung giữa các chức năng 19](#_Toc87232671)

[a. Client 19](#_Toc87232672)

[b. Server 19](#_Toc87232673)

[4. APPS/PROCESSES CONTROLLER 19](#_Toc87232674)

[a. Client 19](#_Toc87232675)

[b. Server 20](#_Toc87232676)

[5. FTP CONTROLLER 20](#_Toc87232677)

[a. Client 20](#_Toc87232678)

[b. Server 21](#_Toc87232679)

[6. KEYBOARD CONTROLLER 21](#_Toc87232680)

[7. MAC ADDRESS 21](#_Toc87232681)

[a. Client 21](#_Toc87232682)

[b. Server 21](#_Toc87232683)

[8. POWER CONTROLLER 22](#_Toc87232684)

[a. Client 22](#_Toc87232685)

[b. Server 22](#_Toc87232686)

[9. STREAMING CONTROLLER 22](#_Toc87232687)

[a. Client 22](#_Toc87232688)

[b. Server 22](#_Toc87232689)

[10. REGISTRY CONTROLLER 22](#_Toc87232690)

[a. Client 22](#_Toc87232691)

[b. Server 23](#_Toc87232692)

[11. Disconnect 23](#_Toc87232693)

[a. Client 23](#_Toc87232694)

[b. Server: 23](#_Toc87232695)

[VI. Tài liệu tham khảo 23](#_Toc87232696)

# Môi trường làm việc, ngôn ngữ, thư viện hỗ trợ

## Môi trường làm việc:

Visual Studio Code

## Ngôn ngữ:

Python

## Thư viện hỗ trợ:

* Truyền nhận dữ liệu: Socket
* Giao diện đồ họa: Tkinter, PIL
* App, Process: subprocess
* FTP: os, signal, psutil, shutil. win32com.client
* Keyboard: ctypes
* MAC: psutil
* Power: ctypes, os
* Streaming: cv2, pickle
* Registry: winreg
* Non-type: struct, threading, json

# Mô tả các file trong source code

## Thư mục Client

* Client.py: code giao diện của client, các hàm truyền nhận và chức năng.
* mySocket.py: Chứa class MySocket, kế thừa socket.socket, tối ưu trong việc truyền nhận dữ liệu.
* File clientIcon: File icon của ứng dụng
* Các file ảnh png: Các hình ảnh hỗ trợ giao diện

## Thư mục Server

* Server.py: file python chính, chứa class Server, gọi các class khác thực thi chức năng.
* Các file .py còn lại: Chứa các class hỗ trợ
* appControler.py: class
* processControler.py
* FTPControler.py
* KeyboardControler.py
* MACControler.py
* powerControler.py
* streamingControler.py
* registryControler.py
* mySocket.py

# Hướng dẫn sử dụng và tính năng

## Kết nối Client-Server

Trên máy chủ server, bấm vào nút "Open server".

Graphical user interface, application

Description automatically generated

Trên giao diện Client, Có các tab APPS CONTROLLER, PROCESSES CONTROLLER, FTP CONTROLLER, KEYBOARD CONTROLLER, MAC ADDRESS, POWER CONTROLLER, STREAMING CONTROLLER, REGISTRY CONTROLLER, khu vực nhập IP, nút Connect và nút Disconnect. Các tab và nút Disconnect đã bị vô hiệu hóa cho chưa thiết lập kết nối.

Graphical user interface, text

Description automatically generated

Trên máy chủ client, nhập địa chỉa IP của máy chủ server trong cổng nhập "Enter IP address, sau đó bấm vào Connect. Kết nối Client-Server được thiết lập. Các Tab và nút Disconnect được mở khóa, nút Connect đã bị vô hiệu hóa.

## Tab APPS CONTROLLER

Chúng tôi dùng thư viện subprocess để lấy kết quả các apps đang chạy thu được khi chạy lệnh trên powershell như lấy processes, yêu cầu các processes này phải có thuộc tính MainWindowTitle thì mới xem là một app.

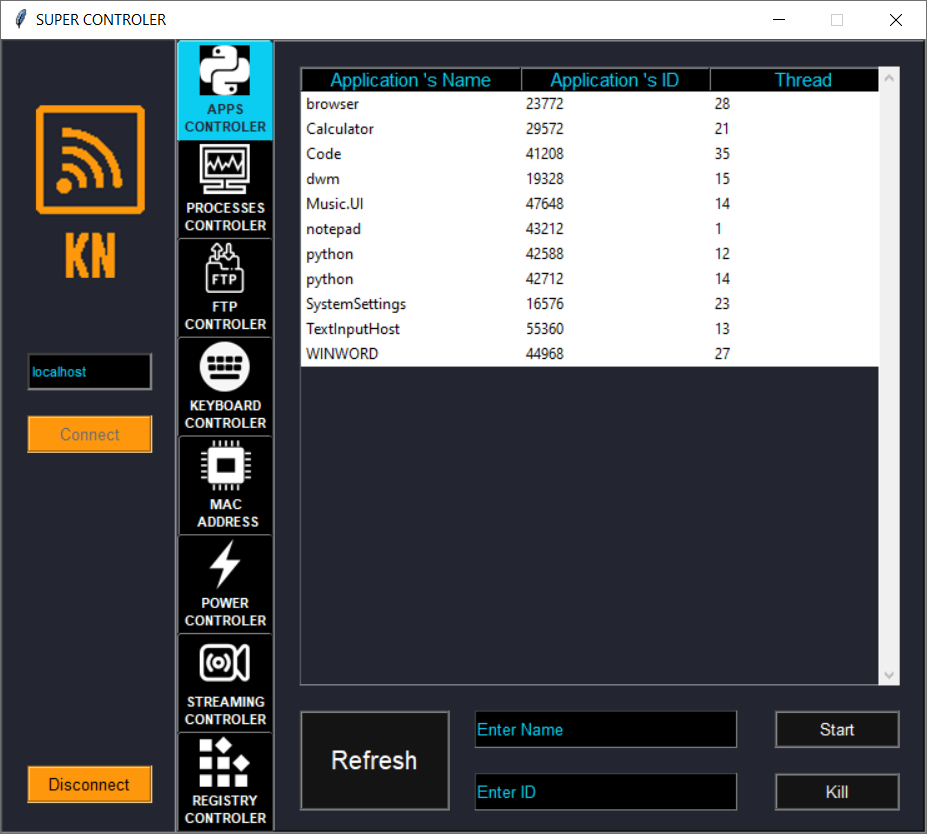
Khi kết nối được thiết lập, tab APPS CONTROLLER được mở mặc định vì là tab đầu tiên, hoặc người dùng bấm vào icon của tab APP CONTROLLER để mở.

A screenshot of a computer

Description automatically generated with medium confidence

### Refresh

Trên tab của APPS CONTROLLER, bấm vào nút Refresh, thông tin về các App đang chạy sẽ xuất hiện.



### Start

Trên tab của APPS CONTROLLER, nhập tên App muốn khởi chạy vào cổng nhập Enter Name, bấm vào nút Start. Nếu trên máy chủ Server tồn tại App, App sẽ được khởi chạy.

A screenshot of a computer

Description automatically generated with medium confidence

### Kill

Có 2 cách để diệt 1 App đang chạy:

* Trên tab của APPS CONTROLLER, nhập ID App muốn kill vào cổng nhập Enter ID, bấm vào nút Kill. Nếu trên máy chủ Server tồn tại App đang chạy có ID vừa nhập, App sẽ bị diệt.
* Trên bảng hiện thị thông tin của các App, nhấn chuột phải vào App muốn diệt, chọn Kill, App sẽ bị diệt.



## Tab PROCESSES CONTROLLER

Chúng tôi dùng thư viện subprocess để lấy kết quả các processes đang chạy thu được khi chạy lệnh trên powershell.

Khi kết nối được thiết lập, người dùng bấn vào icon của tab PROCESSES CONTROLLER để mở.



### Refresh

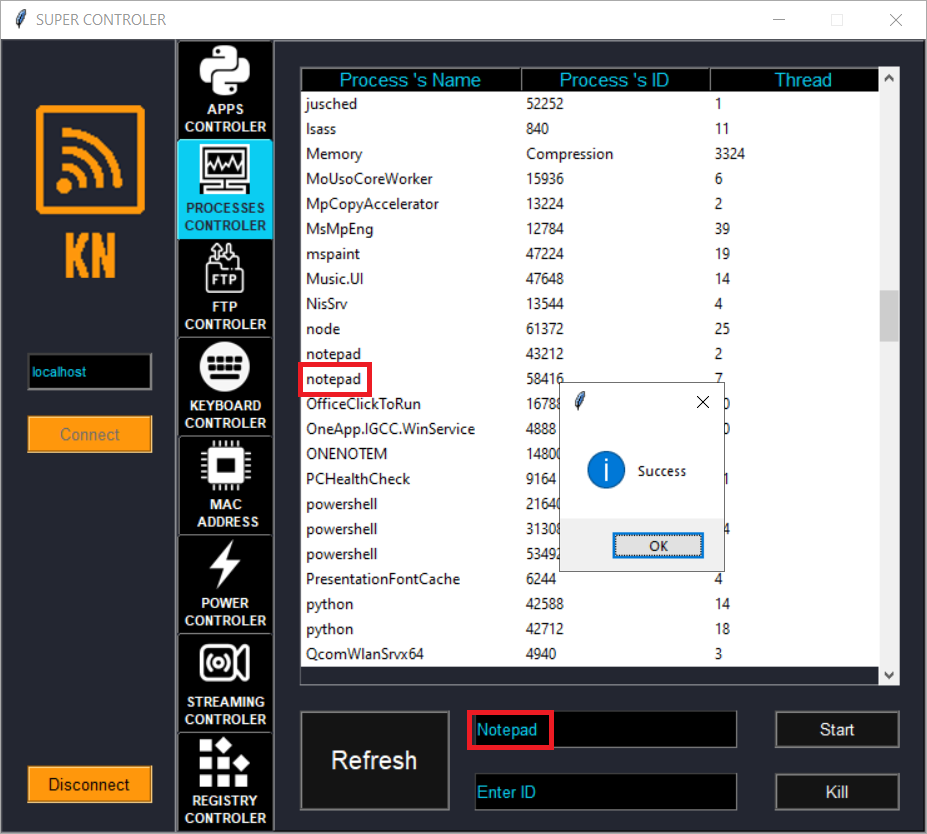
Trên tab của PROCESSES CONTROLLER, bấm vào nút Refresh, thông tin về các Process đang chạy sẽ xuất hiện.

A screenshot of a computer

Description automatically generated with medium confidence

### Start

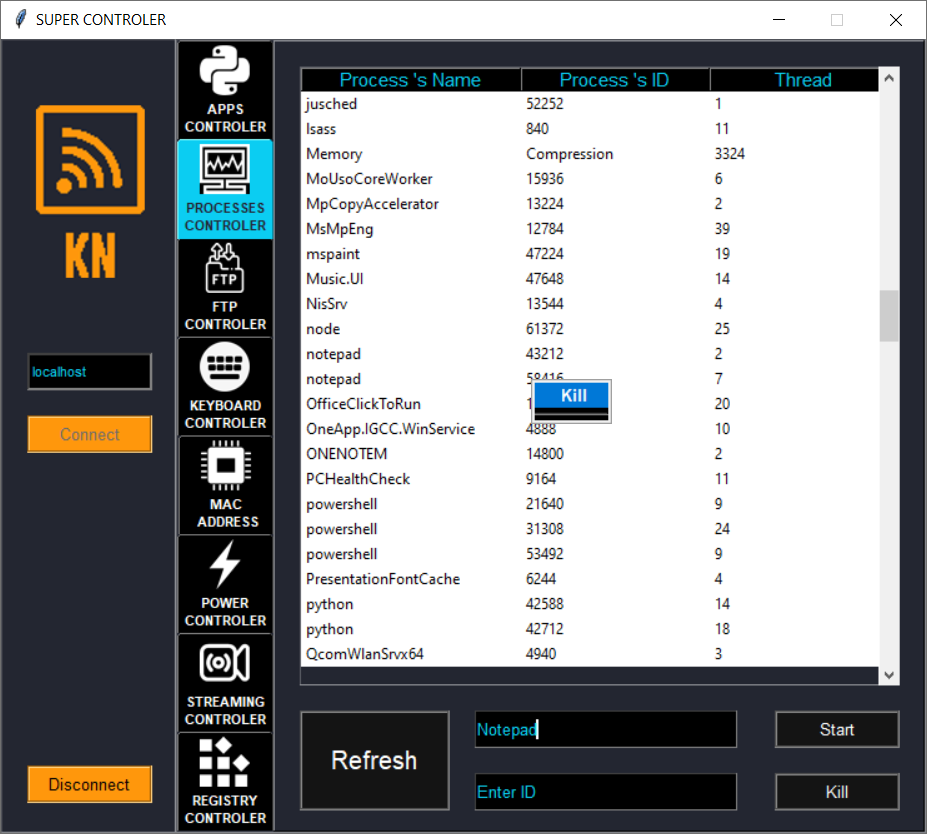
Trên tab của PROCESSES CONTROLLER, nhập tên Process muốn khởi chạy vào cổng nhập Enter Name, bấm vào nút Start. Nếu trên máy chủ Server tồn tại Process, Process sẽ được khởi chạy.



### Kill

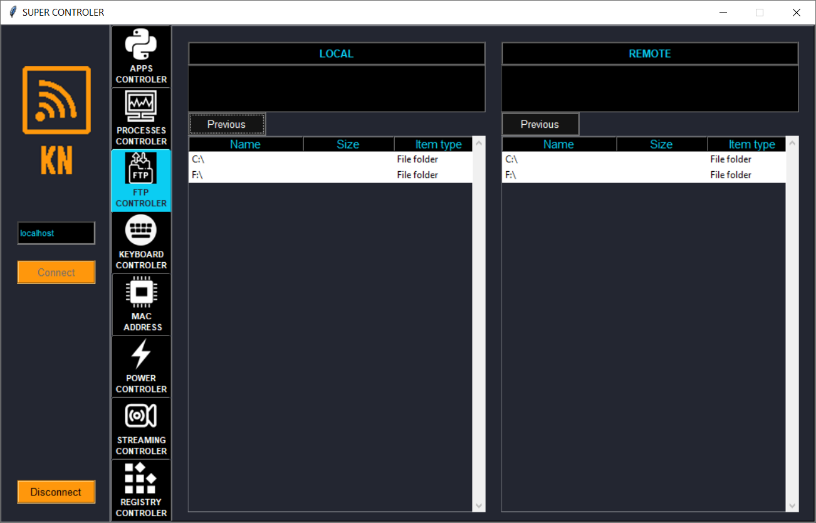
Có 2 cách để diệt 1 Process đang chạy:

* Trên tab của PROCESSES CONTROLLER, nhập ID Process muốn kill vào cổng nhập Enter ID, bấm vào nút Kill. Nếu trên máy chủ Server tồn tại Process đang chạy có ID vừa nhập, Process sẽ bị diệt.
* Trên bảng hiện thị thông tin của các Process, nhấn chuột phải vào Process muốn diệt, chọn Kill, Process sẽ bị diệt.



## Tab FTP CONTROLLER

Khi kết nối được thiết lập, người dùng bấn vào icon của tab PROCESSES CONTROLLER để mở.

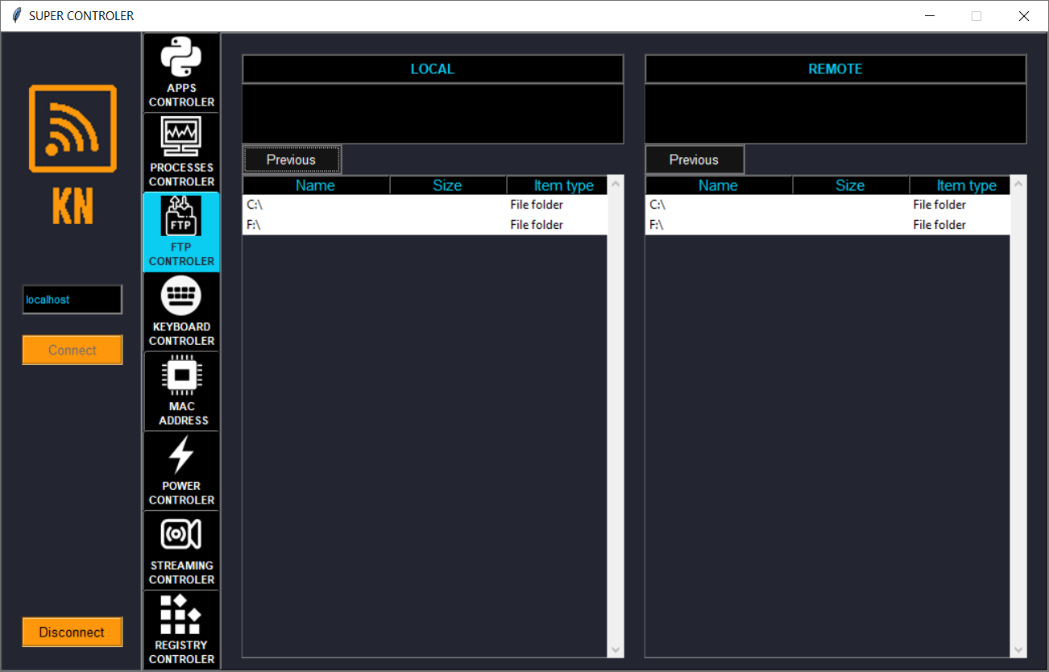


### Hiển thị đường dẫn

Trên tab FTP CONTROLLER, sẽ có 2 cột hiển thị thông tin, 1 bên là của Client, 1 bên là của Server.

Cấu trúc hiển thị từ trên xuống dưới là:

* Nhãn cho biết cột Client hay Server
* Hộp hiển thị text cho biết đường dẫn
* Nút Previous để quay về cây thư mục cha
* Cây hiển thị các thư mục con trong đường dẫn



### Delete Folder(File) trên Client

Nhấn chuột phải vào thư mục muốn delete trên cây hiển thị file của Client, chọn Delete, thư mục sẽ bị xóa.

Graphical user interface, text, website

Description automatically generated

### Send Folder(File) từ Client tới Server

Chọn đường dẫn file đích gửi tới trên Server. Nhấn chuột phải vào thư mục muốn send trên cây hiển thị file của Client, chọn send, thư mục sẽ được copy sang thư mục đích đã chọn trên máy chủ Server.

Graphical user interface, text, application

Description automatically generated

Trường hợp trùng tên thư mục, sẽ xuất hiện hộp thoại, có các lựa chọn để xử lý để overwrite, rename hoặc cancel.

### Delete Folder(File) trên Server

Nhấn chuột phải vào thư mục muốn delete trên cây hiển thị file của Server, chọn Delete, thư mục sẽ bị xóa.

Graphical user interface, text

Description automatically generated

### Lấy Folder(File) từ Server sang Client

Chọn đường dẫn file đích muốn lưu trên Client. Nhấn chuột phải vào thư mục muốn lấy trên cây hiển thị file của Server, chọn Get, thư mục sẽ được copy sang thư mục đích đã chọn trên máy chủ Client.

Graphical user interface, text, application, chat or text message

Description automatically generated

Trường hợp trùng tên thư mục, sẽ xuất hiện hộp thoại, có các lựa chọn để xử lý để overwrite, rename hoặc cancel.

## Tab KEYBOARD CONTROLLER

Khi kết nối được thiết lập, người dùng bấn vào icon của tab KEYBOARD CONTROLLER để mở.



### Lock

Nhấn Lock, bàn phím trên máy chủ Server sẽ bị khóa và sẽ không nhập được. Sau khi nhấn thì nút Lock sẽ chuyển thành Unlock. Nhấn Unlock để mở khóa bàn phím.

### Hook

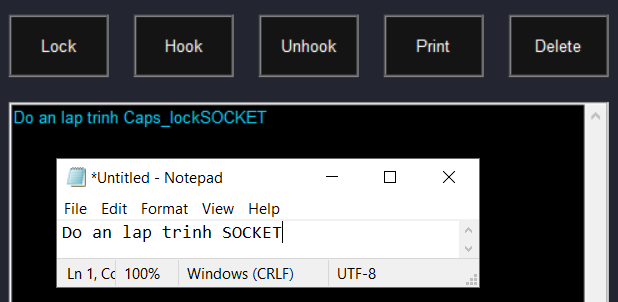
Bấm vào nút Hook, Client sẽ bắt đầu ghi lại thao tác bàn phím trên Server.

### Unhook

Bấm vào nút Unhook, Client sẽ ngừng ghi lại thao tác bàn phím trên Server.

### Print

Bấm vào nút Print, Client sẽ in ra thao tác bàn phím đã ghi lại được từ Server từ thời điểm sau khi print trước đó.

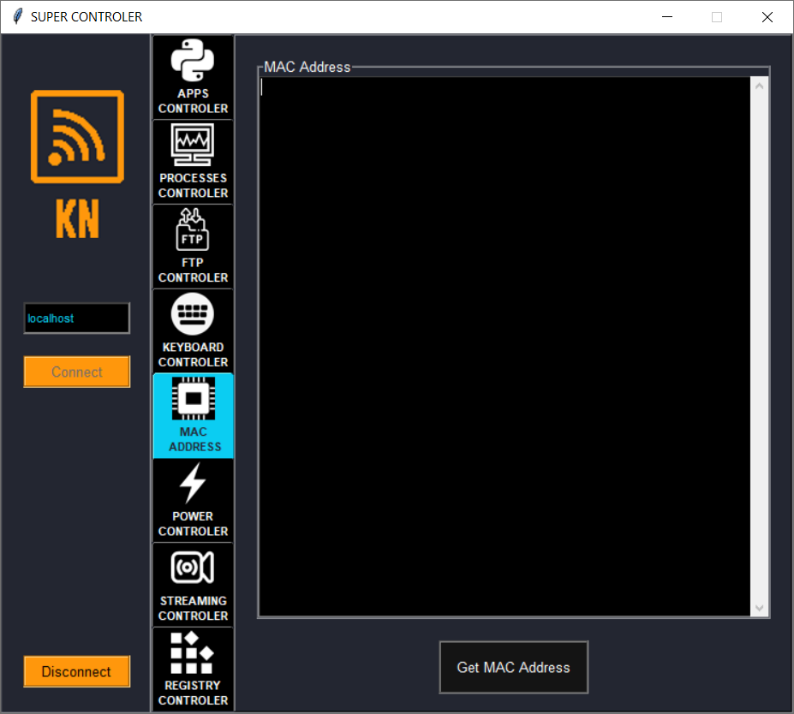


### Delete

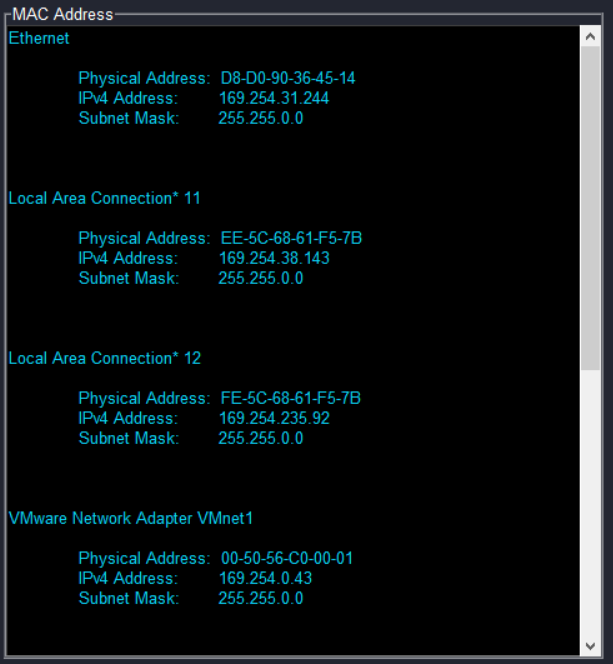
Bấm vào nút Delete sẽ xóa đi tất cảc trình thao tác bàn phím được hiển thị.

## Tab MAC CONTROLLER

Khi kết nối được thiết lập, người dùng bấn vào icon của tab MAC ADDRESS để mở.

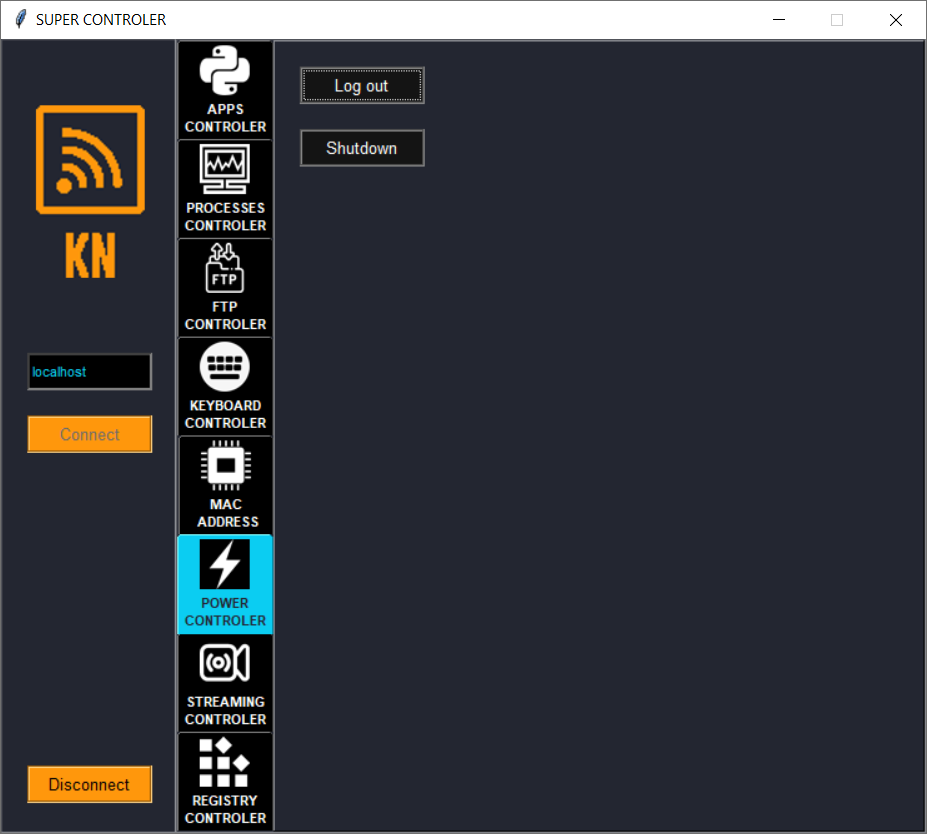


Bấm vào nút Get MAC Address, sẽ hiện thị thông tin các địa chỉ MAC trên máy chủ server.



## Tab POWER CONTROLLER

Khi kết nối được thiết lập, người dùng bấn vào icon của tab POWER CONTROLLER để mở.



### Log out

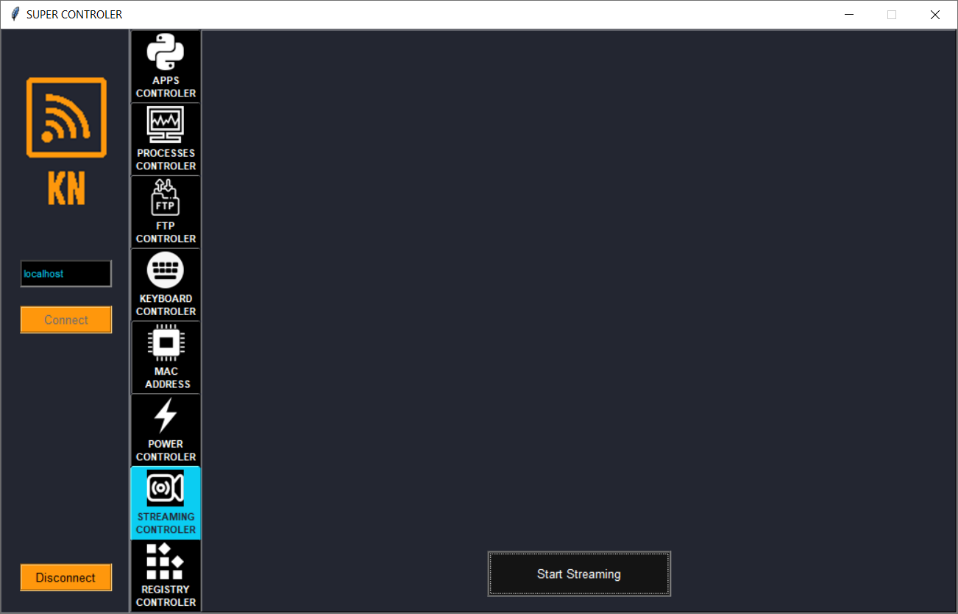
Bấm vào nút Log out, máy chủ Server sẽ bị khóa và yêu cầu đăng nhập lại Window

### Shutdown

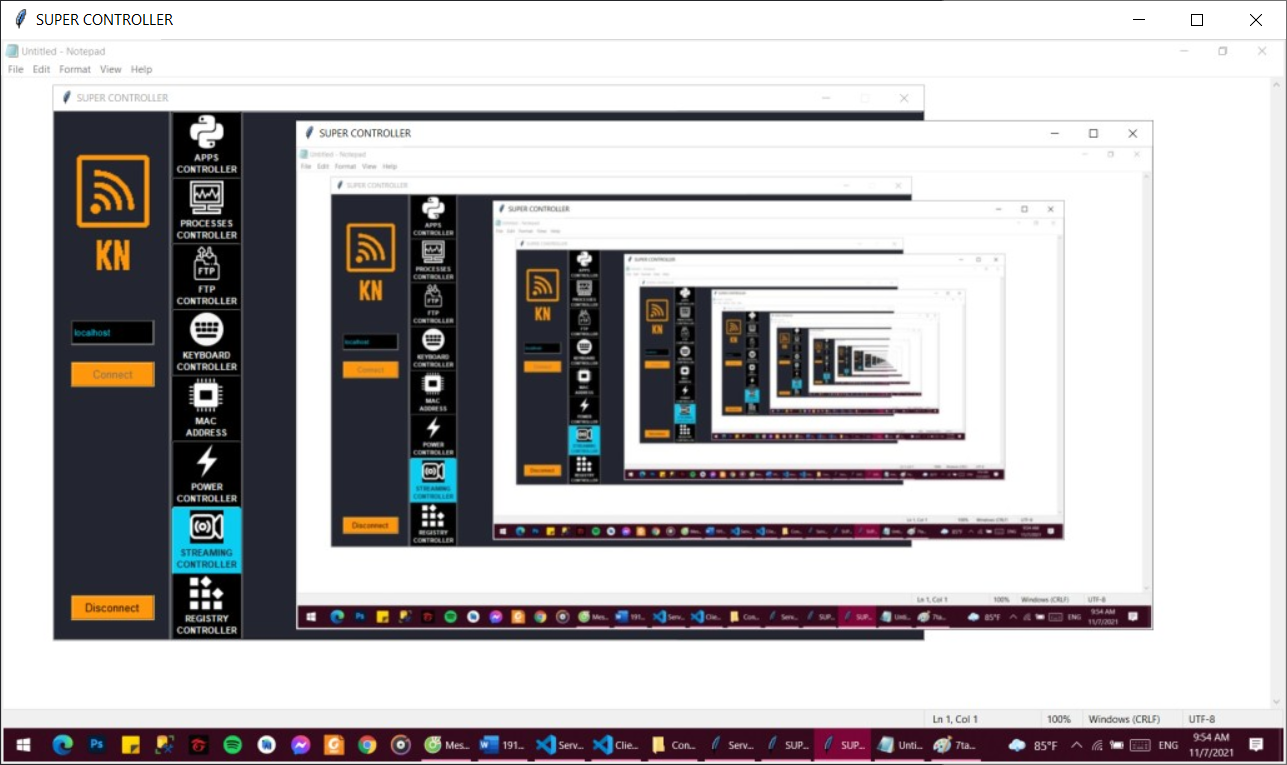
Bấm vào nút Shutdown, máy chủ Server sẽ tắt sau 60 giây tính từ lúc bấm.

## Tab STREAMING CONTROLLER

Khi kết nối được thiết lập, người dùng bấn vào icon của tab STREAMING CONTROLLER để mở.



Bấm vào nút Start Streaming, trên Client sẽ khởi tạo 1 cửa sổ mới và hiển thị trực tiếp màn hình của máy chủ Server. Người dùng có thể thao tác với các chức năng khác vừa có thẻ xem live màn hình.



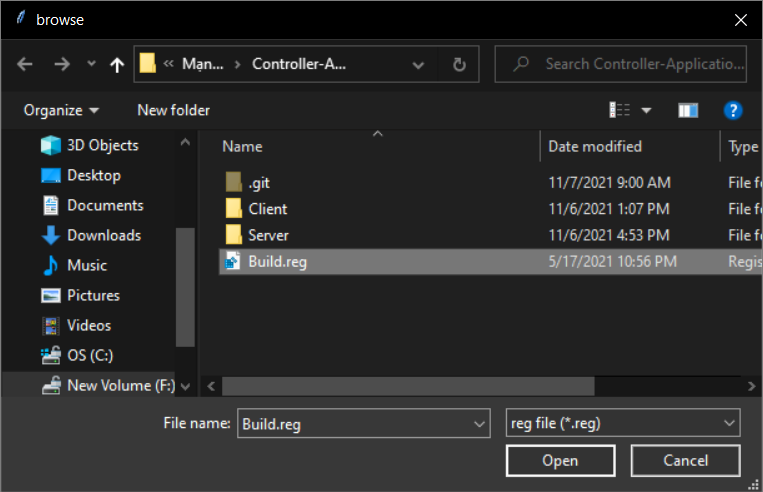
## Tab REGISTRY CONTROLLER

Khi kết nối được thiết lập, người dùng bấn vào icon của tab REGISTRY CONTROLLER để mở.



### Sửa bằng File Registry

Bấm vào nút Browse, hộp thoại Browse xuất hiện. Chọn file có định dạng .reg hoặc file có định dạng khác nhưng nội dung tương tự file .reg.



Nhấn nút Open. Nội dung của file được chọn sẽ xuất hiện trong khung File 's content.

Graphical user interface, text, application

Description automatically generated

Có thể chỉnh sửa nội dung trong khung. Sau khi chỉnh sửa xong, nhấn nút Send content. Sau khi Server nhận được file, Client sẽ nhận được thông báo nội dung file .reg thực thi thành công hay thất bại trên Server.

### Sửa trực tiếp

Nhập đường dẫn tới file registry cần chỉnh sửa trên Server.

Graphical user interface, application

Description automatically generated

Chọn chức năng để thao tác với file registry. Có các chứng năng như sau:

* Lấy giá trị: Nhập tên.
* Sửa giá trị: Nhập tên, giá trị, kiểu dữ liệu.
* Xóa giá trị: Nhập tên.
* Tạo khóa: Không nhập.
* Xóa khóa: Không nhập.

Nhấn nút Send. Sau khi nhận được các giá trị cần thiết, Server sẽ gửi cho Client thông báo chỉnh sửa file Registry thành công hay thất bại. Kết quả sẽ được hiển thị trong khung ở dưới cùng. Nhấn nút Delete để xóa khung kết quả.

## Disconnect

Khi kết nối được thiết lập, người dùng bấm nút Disconnect, kết nối sẽ bị hủy, các tab và nút Disconnect sẽ bị vô hiệu hóa, nút Connect sẽ mở khóa, Server trở về trang thái ban đầu.

# Mô tả mã nguồn đồ án

1. MySocket.py

* Class MySocket kế thừ từ socket.socket để truyền nhận tín hiệu chủ yếu trong sản phẩm
* Các hàm *send, recv, close, accept* được overwrite lại để đảm bảo không bị deadlock khi thao tác với socket
* Hàm *send* sẽ gửi kích thước (8 bytes) của dữ liệu được gửi qua trước, sau đó mới gửi dữ liệu
* Hàm *recv* nhận kích thước (8 bytes) và nhận từng phần của dữ liệu cho đến khi nhận hết với kích thước

## Connect

### Client

Nhấn vào nút Connect trên giao diện Client:

* Client sẽ gọi hàm *butConnectClick(self, event = None)*
* Client sẽ thử kết nối tới địa chỉ IP do người dùng nhập vào
* Nếu địa chỉ IP của Server đang chờ kết nối thì kết nối thành công và thông báo lại cho người dùng

### Server

Nhấn vào chức năng Open server của Server:

* Server gọi hàm *buttonClick(self)* để tạo socket và chờ kết nối

## Message chung giữa các chức năng

### Client

* Hàm *on\_tab\_change(self,event=None)*: Khi nhấn vào 1 tab bất kì thì hàm on\_tab\_change sẽ được gọi. Hàm này có nhiệm vụ gửi đi message tương ứng với tab được chọn, setup các biến cần thiết và resize cửa sổ. Kể từ lần gọi hàm lần 2, Client sẽ gửi message “quit” trước khi gửi message về chức năng, mục đích là báo đổi tab

### Server

* Khi trạng thái kết nối được thiết lập, Server khởi tạo các đối tượng dựa trên các class hỗ trợ và chờ message từ Client. Tùy theo message nhận được thì mà gọi hàm *startListening(self)* của class tương ứng.

Ví dụ message là “KEYBOARD” thì gọi hàm *KeyboardController.startListening()* (trong file *KeyboardController*)

## APPS/PROCESSES CONTROLLER

### Client

Nhấn nút Refresh: gọi hàm *butRefreshClick(self)*

* Kiểm tra kết nối, gửi message "view"
* Nhận và hiển thị danh sách App/Process từ server

Nhấn nút Start: hàm *butStartClick(self)*

* Kiểm tra kết nối, gửi message "start"
* Gửi tên App/Process, nhận và hiển thị kết quả trả về

Nhấn nút Kill/chuột phải chọn Kill: hàm *butKillClick(self)/killRightClick(self)*

* Kiểm tra kết nối, gửi message "kill"
* Gửi ID App/Process, nhận và hiển thị kết quả trả về

### Server

Server nhận message: **Hàm xử lý trong file *appController/processController***

* "view": Lấy thông tin app/process bằng powershell: *subprocess.check\_output*
* "kill": Kill app/process bằng *os.kill*
* "start": Start app/process bằng *os.startfile*
* Khác: Thoát vòng lặp và kết thúc hàm

## FTP CONTROLLER

### Client

Double click folder của Client: hàm *clientOnDoubleClick(self, event = None)*

Double click folder của Server: hàm *serverOnDoubleClick(self, event = None)*

* Kiểm tra kết nối, gửi message “view”
* Gửi đường dẫn file, nhận và hiển thị danh sách file trả về

Nhấn nút Previous của Client: hàm *butClientPreviousPathClick(self, event = None)*

Nhấn nút Previous của Server: hàm *butServerPreviousPathClick(self, event = None)*

* Kiểm tra kết nối, gửi message “back”
* Gửi đường dẫn file, nhận và hiển thị danh sách file trả về

Chuột phải, chọn Copy trên Client: *copyToServer(self)*

* Kiểm tra kết nối, gửi message “copy2server”
* Gửi toàn bộ nội dung thư mục, làm mới cửa sổ Server

Chuột phải, chọn Delete trên Client: *deleteClientFile(self)*

* Xóa toàn bộ thư mục, làm mới của sổ Client

Chuột phải, chọn Get trên Server: *copyToServer(self)*

* Kiểm tra kết nối, gửi message “copy2client”
* Gửi đường dẫn thư mục, nhận nội dung thư mục, làm mới cửa sổ Client

Chuột phải, chọn Delete trên Server: *deleteServerFile(self)*

* Kiểm tra kết nối, gửi message “delete”
* Gửi đường dẫn thư mục, làm mới cửa sổ Server

### Server

Server nhận message: **Hàm xử lý trong file *ftpController***

* "view"/"back": Lấy thông tin folder bằng *os.walk()*
* "copy2server"/"copy2client": Gửi và nhận file/folder từng phần bằng cách khởi tạo thêm 1 socket để gửi/nhận data (gửi folder sử dụng thêm *os.walk*)
* "delete": Xóa file/folder bằng hàm *os.remove/shutil.rmtree*
* Khác: Thoát vòng lặp và kết thúc hàm
* Hàm hỗ trợ: Lấy thông tin ổ cứng bằng *psutil.disk\_partitions*, lấy thông tin về file bằng hàm *getFileMetadata* của class

## KEYBOARD CONTROLLER

1. Client

* Nhấn nút Lock/Unlock: Gửi message “lock”/”unlock” hàm *butLockClick(self)*
* Nhấn nút Hook/Unhock: Gửi message “hook”/”unhook” bằng hàm *butHookClick(self)/butUnhookClick(self)*
* Nhấn nút Print: Gửi message “Print”, nhận chuỗi kí tự và hiển thị lên cửa sổ hàm *butPrint(click)*
* Nhấn nút Delete: Làm mới của sổ hàm *butDelClick4(self)*

1. Server

Server nhận message: **Hàm xử lý trong file *keyboardController***

* "lock": Khóa bàn phím và chuột *windll.user32.BlockInput(True)*
* "unlock": Mở khóa bàn phím và chuột *windll.user32.BlockInput(False)*
* "hook": Tạo thread để bắt phím, gọi *KeyLog.KeyHook().installHookProc*
* "unhook": Gọi hàm *KeyLog.KeyHook().unistallHookProc()*
* "print": Gửi chuỗi ký tự bắt được và unhook
* Khác: Thoát vòng lặp, unhook và kết thúc hàm

## MAC ADDRESS

### Client

Nhấn nút Get MAC Address: hàm *butGetMACClick(self)*

* Gửi message “macaddress”
* Nhận về danh sách các MAC address, hiển thị trong cửa sổ

1. Server

Server nhận message: **Hàm xử lý trong file *macAddressController***

* “macaddress”: Lấy danh sách địa chỉ MAC bằng *psutil.net\_if\_addrs().items()*
* Khác: Thoát vòng lặp và kết thúc hàm

## POWER CONTROLLER

### Client

* Nhấn nút Log out: gửi message “logout” hàm *butLogOutClick(self)*
* Nhấn nút Shutdown gửi message “shutdown” *butShutDownClick(self)*

### Server

Server nhận message: **Hàm xử lý trong file *powerController***

* “logout”: Logout bằng *windll.user32.ExitWindowsEx(0, 1)*
* “shutdown”: Shutdown bằng *os.system("shutdown /s /t 60")*
* Khác: Thoát vòng lặp và kết thúc hàm

## STREAMING CONTROLLER

### Client

Nhấn nút Start Streaming: *butStartRecording(self)*

* Tạo cửa sổ hiển thị mới, gọi class Streaming Server
* Gửi message “stream”
* Nếu thoát cửa sổ sẽ gửi message “stop”

### Server

Server nhận message: **Hàm xử lý trong file *streamingClient***

* “stream”: Stream bằng class *StreamingClient* (mã nguồn của thư viện *vidstream*)
* “stop”: Đóng socket, kết thúc stream
* Khác: Thoát vòng lặp và kết thúc hàm

## REGISTRY CONTROLLER

### Client

Nhấn nút Send content: hàm *butSend1Click(self)*

* Gửi message “reg”
* Gửi nội dung file reg đã chọn
* Nhận kết quả trả về và hiện thị dưới dạng thông báo

Nhấn nút Send: hàm *butSend2Click(self)*

* Gửi message “edit”
* Gửi thông tin về chức năng, đường dẫn, tên, giá trị, kiểu dữ liệu
* Nhận kết quả trả về và hiện thị trong cửa sổ

### Server

Server nhận message: **Hàm xử lý trong file *registryClient***

* “reg”: Nhận file reg và thêm bằng *os.system("reg import fileReg.reg")*
* “edit”: Tùy vào chức năng mà xử lý bằng thư viện *winreg*
* Khác: Thoát vòng lặp và kết thúc hàm

## Disconnect

### Client

Hàm *butDisconnectClick(self)*

* Client gửi message “quit” để thoát khỏi chức năng tab hiện tại.
* Client gửi message “EXIT”
* Đóng socket

### Server:

Server nhận message:

“quit”: Thoát chức năng hiện tại

“EXIT”: Đóng socket, trở về trạng thái ban đầu.

# Tài liệu tham khảo

*[1] Thư viện để stream màn hình:* [*https://github.com/NeuralNine/vidstream*](https://github.com/NeuralNine/vidstream)

*[2] Code gửi nhận folder:* [*https://stackoverflow.com/questions/47539028/transfer-contents-of-a-folder-over-network-by-python*](https://stackoverflow.com/questions/47539028/transfer-contents-of-a-folder-over-network-by-python)

*[3] Chức năng keylogger:* [*https://gist.github.com/andreafortuna/66565082342076908a955cc4b041c2fc*](https://gist.github.com/andreafortuna/66565082342076908a955cc4b041c2fc)