

MẠNG MÁY TÍNH

BÁO CÁO ĐỒ ÁN MÔN MẠNG MÁY TÍNH

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

22127414 - Từ Chí Tiến

22127471 - Ngô Hải Bằng

22127083 - Cao Hữu Khương Duy



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

MỤC LỤC

Chủ đề

1. Tổng quan	3
1.1. Thông tin nhóm.....	3
1.2. Thông tin đồ án.....	3
2. Ý tưởng về việc kết nối máy Client và máy Server.....	4
3. Chi tiết về các tính năng.....	5
3.1. Kết nối Client và Server	5
3.2. Chụp ảnh màn hình	6
.....	6
3.3. Điều khiển các tiến trình đang chạy nền.....	7
3.4. Điều khiển các ứng dụng đang hoạt động	8
3.5. KeyStroke.....	9
3.6. Tắt máy	10
3.7. Ngắt kết nối Server	10
4. Đánh giá	11
4.1. Đánh giá mức độ hoàn thành	11
4.2. Đánh giá tỉ lệ đóng góp	11

1. Tổng quan

1.1. Thông tin nhóm

MSSV	Họ tên	Email	Vai trò
22127414	Từ Chí Tiến	22127414@student.hcmus.edu.vn	Thành viên
22127471	Ngô Hải Bằng	22127471@student.hcmus.edu.vn	Thành viên
22127083	Cao Hữu Khương Duy	22127083@student.hcmus.edu.vn	Nhóm trưởng

1.2. Thông tin đồ án

Tên đồ án	
Công cụ hướng dẫn	Source Code Python Demo (cre: thầy Đỗ Hoàng Cường)
	Video hướng dẫn (cre: thầy Đỗ Hoàng Cường)
	Tài liệu của môn học Mạng máy tính
	Google, Chat GPT, ...

Báo cáo sau đây trình bày chi tiết về Đồ án Lập trình Socket, ứng dụng được thực hiện bằng ngôn ngữ lập trình Python, tạo kết nối giữa các thiết bị qua mạng và cách trao đổi dữ liệu thông qua giao thức TCP/IP. Hơn nữa, việc thực hiện việc xử lý đồng thời và đồng bộ hóa cũng là một khía cạnh quan trọng trong việc phát triển ứng dụng mạng.

Ứng dụng do nhóm tạo ra sẽ thực hiện việc tạo ra 1 máy chủ (Server) và 1 máy khách (Client), máy Client sẽ nhập địa chỉ IP của mạng ở máy Server và kết nối với nhau. Qua đó, máy Client có thể thực hiện một số tính năng như: chụp màn hình, xem các ứng dụng đang hoạt động, KeyStroke, tắt máy,... đối với máy Server.

2. Ý tưởng về việc kết nối máy Client và máy Server

Trong đồ án này, việc kết nối máy Client và máy Server được thực hiện bằng cách sử dụng giao thức TCP/IP và các khái niệm về Socket, tạo ra một cơ chế cho phép 2 máy tính thực hiện việc truyền tải, trao đổi, chia sẻ và tương tác với nhau thông qua mạng.

Máy Client, trong trường hợp này là điểm khởi đầu của ứng dụng. Khi máy Client muốn gửi yêu cầu đến máy Server, nó sử dụng Socket để thiết lập kết nối với máy Server và truyền tải dữ liệu. Máy Server, được lập trình để lắng nghe các yêu cầu từ máy Client thông qua Socket tương ứng, sau đó sẽ xử lý yêu cầu, thực hiện các tác vụ hay trả về các dữ liệu từ máy Client gửi lên.

Mô hình kết nối Client-Server mang lại tính phân chia rõ ràng giữa vai trò và chức năng của các máy trong mạng. Máy Client sẽ gửi yêu cầu và nhận dữ liệu, trong khi máy Server chịu trách nhiệm xử lý yêu cầu và cung cấp dữ liệu tương ứng.

Máy Client sẽ gửi những tín hiệu từ input của người dùng, sau đó máy Server sẽ lần lượt nhận, xử lý và trả kết quả tới client. Client sẽ nhận lại output tương ứng và in ra màn hình.

Việc kết nối máy Client và máy Server thông qua giao thức TCP/IP và sử dụng Socket là bước quan trọng trong việc xây dựng hệ thống mạng hoạt động tốt. Điều này tạo khả năng truyền tải dữ liệu đáng tin cậy và hiệu quả, cũng như tạo nên nền tảng cho sự tương tác và trao đổi thông tin giữa các thiết bị mạng.

3. Chi tiết về các tính năng

3.1. Kết nối Client và Server



Ngay khi máy Client chạy chương trình, một menu dành cho máy Client sẽ được hiện thị lên, trong đó sẽ có một khung để nhập địa chỉ IP máy Server cần kết nối và một Nút “Connect” (như trên).

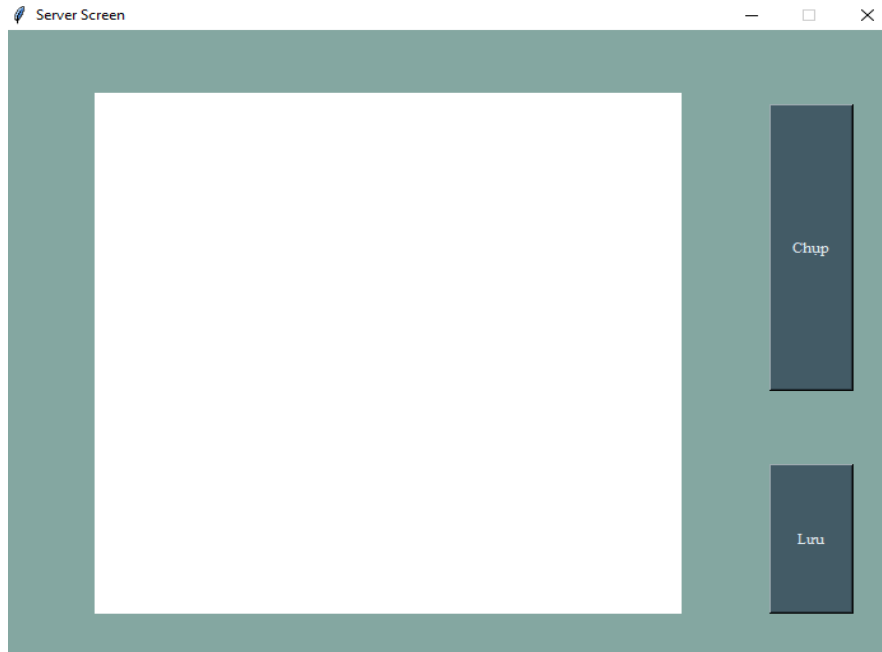
Người dùng sẽ nhập địa chỉ IP của máy Server cần kết nối vào khung này sau đó Click vào button “Connect” để tạo một kết nối Socker đến máy Server.

Lưu ý: người dùng cần nhập đúng địa chỉ của máy Server để kết nối chính xác đến máy đích.

Thông tin của các hàm hỗ trợ:

- + start_client: Khởi động kết nối client tới server, xử lý các lệnh từ server trả về và thực hiện chức năng tương ứng.
- + run_client: Chạy ứng dụng cho đến khi nào nhận được lệnh ‘QUIT’ từ người dùng.
- + handleClientSocket: Xử lý các lệnh gửi từ máy khách, thực thi chức năng tương ứng trên máy chủ.
- + start_server: Bắt đầu máy chủ và lắng nghe các kết nối từ máy khách. Khi có kết nối mới, một thread mới được tạo.

3.2. Chụp ảnh màn hình

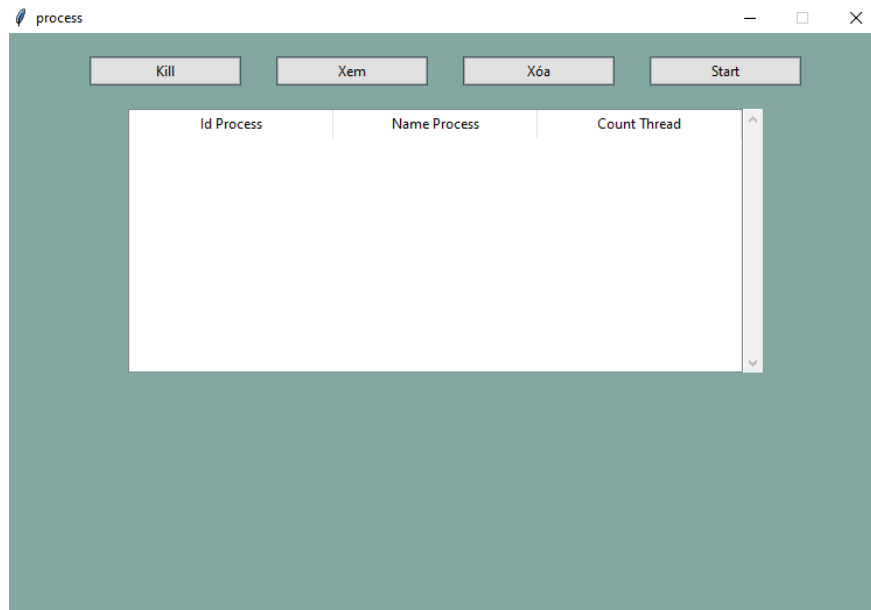


Trong tính năng này, khi click vào button “Chụp” , phía Client sẽ gửi đến Server một yêu cầu và Server sẽ chụp ảnh màn hình, gửi ngược về cho máy Client, hình ảnh sẽ được hiện thị trên khung màu trắng. Nếu click vào button “Luu” máy Client sẽ được mở một bảng File Dialog để chọn nơi lưu hình ảnh chụp được từ máy Server.

Thông tin của các hàm hỗ trợ:

- + saveImage: Mở file để lưu ảnh.
- + readImage: Đọc hình ảnh từ Socket, lưu trữ nó và thông báo rằng hình ảnh đã được nhận thành công.
- + sendScreenShot: Chụp ảnh màn hình, chuyển nó thành dữ liệu byte và gửi nó qua socket.

3.3. Điều khiển các tiến trình đang chạy nền



Bảng hiển thị các Process đang được chạy sẽ được chia thành 3 cột gồm: ID Process, Name Process, Count Thread (số luồng Process sử dụng).

Khi chọn “Start” chương trình sẽ mở 1 cửa sổ, qua đó user có thể nhập ID của Process cần được bắt đầu, sau khi nhập ID Process đó, Process sẽ được bắt đầu trên máy của Server.

Khi chọn “Xem” các process đang chạy sẽ được hiển thị thông qua bảng hiển thị process.

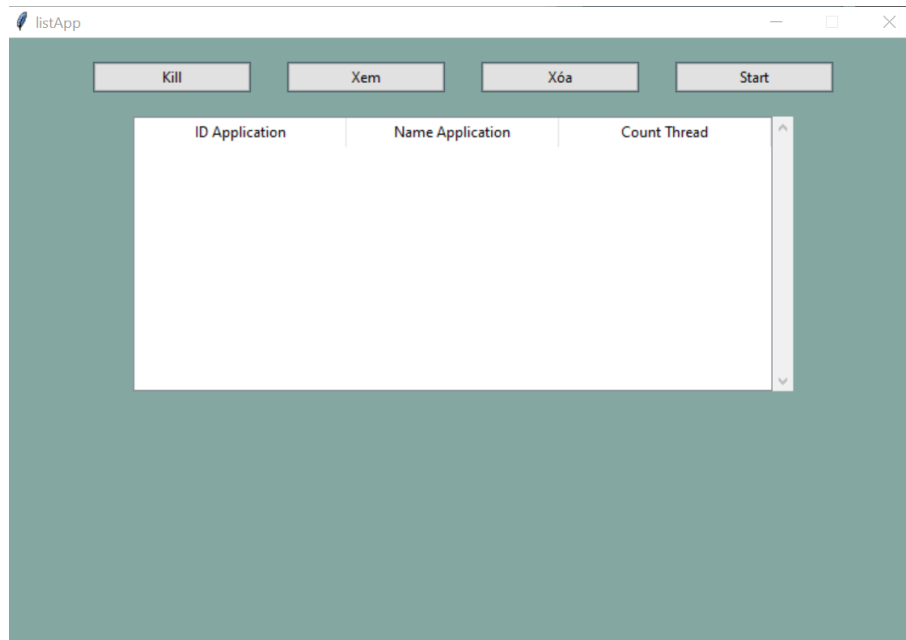
Khi chọn “Kill” chương trình sẽ hiển thị 1 cửa sổ để nhập ID process cần được xóa đi (kết thúc process).

Khi chọn “Xóa” các process đang được hiển thị ở bên trong bảng sẽ được xóa đi (nhưng vẫn tiếp chạy bình thường ở máy Server).

Thông tin của các hàm hỗ trợ:

- + receiveProcess: Nhận và ghi thông tin process vào một tệp.
- + receiveStatus: Nhận trạng thái (kill thành công hay không).
- + killProcess: Kết thúc một process dựa trên PID của nó.
- + listProcess: Lấy thông tin tất cả các process và gửi cho client.

3.4. Điều khiển các ứng dụng đang hoạt động



Tương tự với Process Running, tính năng điều khiển các ứng dụng đang chạy cũng sẽ có một bảng hiển thị các Application đang hoạt động.

Bảng hiển thị các ứng dụng đang chạy cũng bao gồm 3 cột gồm: ID Application, Name Application và Count Thread (số lượng ứng dụng sử dụng).

Khi chọn “Start” chương trình sẽ mở 1 cửa sổ, qua đó user có thể nhập ID của ứng dụng cần được bắt đầu, sau khi nhập ID Application đó, ứng dụng sẽ được bắt đầu trên máy của Server.

Khi chọn “Xem” các ứng dụng đang chạy sẽ được hiển thị thông qua bảng hiển thị.

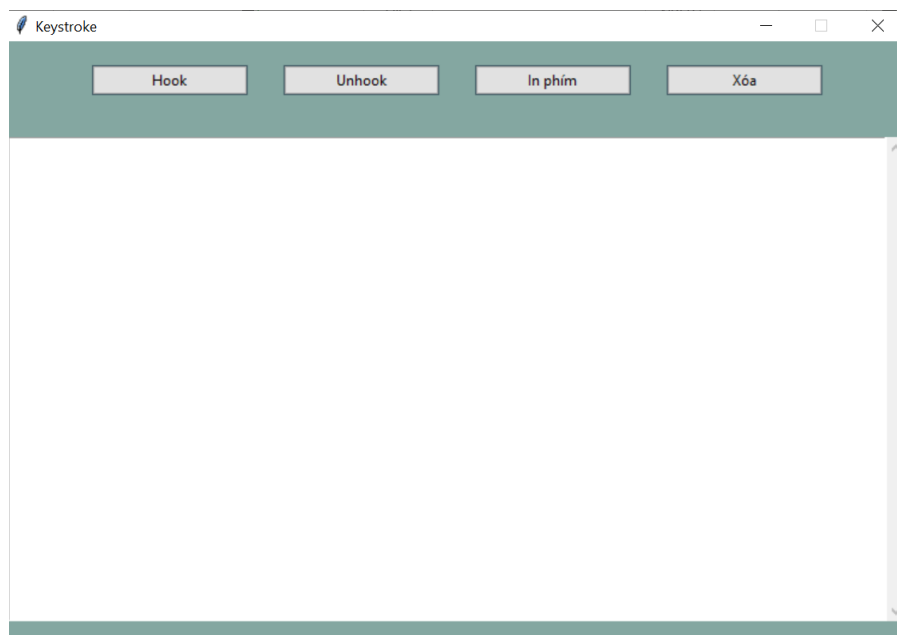
Khi chọn “Kill” chương trình sẽ hiện thị 1 cửa sổ để nhập ID Application cần được xóa đi (dừng hoạt động ứng dụng đó).

Khi chọn “Xóa” các ứng dụng đang được hiển thị ở bên trong bảng sẽ được xóa đi (nhưng vẫn tiếp chạy bình thường ở máy Server).

Thông tin của các hàm hỗ trợ:

- + receiveRunningApp: Nhận danh sách ứng dụng ghi vào tệp.
- + receiveStatus: Nhận trạng thái (kill/op – ok/err).
- + openApp: Mở ứng dụng và gửi trạng thái (thành công hay không).
- + killRunningApp: Tắt ứng dụng đang chạy dựa trên PID và gửi phản hồi.
- + listRunningApp: Lấy danh sách tất cả các ứng dụng đang chạy và gửi cho Client.

3.5. KeyStroke



Tính năng KeyStroke cho phép máy Client bắt các ký tự được gõ bên phía máy Server.

Khi chọn “Hook” Client sẽ gửi 1 tín hiệu cho Server, kể từ khi tín hiệu này được gửi đi, các ký tự bên phía Server gõ sẽ được ghi lại và gửi ngược về cho phía Client.

Khi chọn “Unhook” tín hiệu trên sẽ được ngắt đi, kể từ đó, Client sẽ không nhận được các ký tự mà trên máy Server gõ.

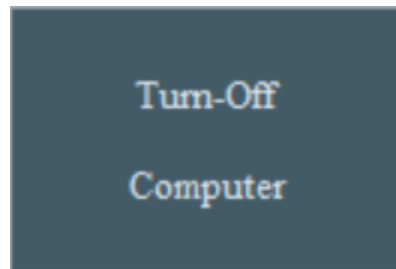
Khi chọn “In phím” các ký tự mà Client bắt được trên máy Server sẽ được hiển thị ra trên một bảng trắng. Qua bảng trắng đó, Client sẽ thấy được các ký tự được bắt trên máy của Server.

Khi chọn “Xóa” các ký tự trên bảng trắng sẽ được xóa đi.

Thông tin của các hàm hỗ trợ:

- + receiveKeylogger: Nhận dữ liệu keylogger từ máy chủ, lưu nó và thông báo rằng dữ liệu đã được nhận thành công.
- + on_key_press: Ghi lại mỗi phím được nhấn và lưu chúng vào một tệp.
- + hook: Khởi tạo một người nghe (listener) để ghi lại mỗi phím được nhấn.
- + deleteKeyloggerFile: Xóa nội dung của tệp Keylogger.
- + sendKeylogger: đọc dữ liệu từ tệp Keylogger, xóa nó và sau đó gửi nội dung đó đến máy khách.
- + startedKeylogger: Khởi động quá trình keylogger bằng cách thiết lập một hook và tạo một thread để in lại phím.

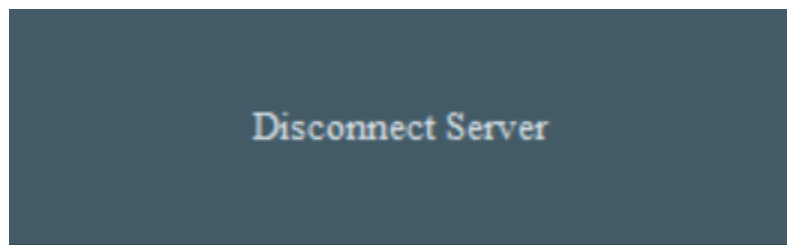
3.6. Tắt máy



Khi máy Client click vào button phía trên (Turn-Off Computer), máy Server sẽ bị tắt (shut-down). Việc này được thực hiện nhờ vào hàm `os.system()` **tác động trực tiếp đến trạng thái máy tính của Server, có thể dẫn đến việc tắt máy Server một cách đột ngột** nên cần phải cẩn thận khi sử dụng tính năng này.

Server chỉ thực hiện một câu lệnh console shutdown /s /t 1 để tắt máy.

3.7. Ngắt kết nối Server



Khi user (máy Client) click vào button phía trên (Disconnect Server), máy Client sẽ gửi đến Server một tín hiệu ngắt kết nối, thực hiện việc đóng kết nối Socket và khôi phục về trạng thái ban đầu (máy Server) để chuẩn bị cho các kết nối tiếp theo.

4. Đánh giá

4.1. Đánh giá mức độ hoàn thành

Chức năng	Mức độ hoàn thành
Mở server	100%
Kết nối Client và Server	100%
Chụp ảnh màn hình và lưu ảnh	100%
Thao tác với các ứng dụng đang chạy	100%
Thao tác với các tiến trình đang chạy	100%
Bắt phím (KeyStroke)	100%
In và xóa phím	100%
Tắt máy server	100%
Ngắt kết nối server	100%
Thoát ứng dụng	100%
Tạo GUI	100%

4.2. Đánh giá tỉ lệ đóng góp

MSSV	Họ tên	Công việc	Tỷ lệ	Ghi chú
22127083	Cao Hữu Khương Duy	Xây dựng các tính năng (Backend) Demo Video	100%	
22127414	Từ Chí Tiến	Xây dựng giao diện người dùng (Frontend) Viết báo cáo	100%	
22127471	Ngô Hải Bằng	Xây dựng giao diện người dùng (Frontend) Viết báo cáo	100%	

5. Video demo

Đường dẫn Youtube đến video:

<https://youtu.be/ly7YuZWnv5Y>