

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN
ĐIỀU KHIỂN MÁY TÍNH TỪ XA BẰNG EMAIL
MÔN HỌC: MẠNG MÁY TÍNH

Giảng viên hướng dẫn: Đỗ Hoàng Cường

Nhóm sinh viên thực hiện:

Cao Trọng Khang 23120132

Khổng Đức Tiến 23120173

Lớp: 23CTT2

Thành phố Hồ Chí Minh, tháng 12 năm 2024

MỤC LỤC

I. Giới thiệu	3
II. Thiết kế đồ án	3
II.1. Luồng hoạt động của ứng dụng.....	3
II.2. Host email	4
II.3. Client (máy điều khiển).....	5
II.3.1. Giao diện app	5
II.3.2. Các chức năng chính	7
II.3.2.1. Kiểm tra mail và gửi mail phản hồi.....	7
II.3.2.2. Gửi yêu cầu của admin cho server và nhận phản hồi.....	7
II.4. Server (máy bị điều khiển).....	8
II.4.1. Nhận yêu cầu và phản hồi cho client	8
II.4.2. Phân tích yêu cầu từ client	9
II.4.3. Xây dựng thông điệp phản hồi	9
II.4.4. Xử lý các lệnh điều khiển.....	10
II.4.4.1. Liệt kê các lệnh điều khiển.....	10
II.4.4.2. Liệt kê, tắt, mở tiến trình (process)	11
II.4.4.3. Liệt kê, tắt, mở ứng dụng (application).....	12
II.4.4.4. Shutdown, Restart	13
II.4.4.5. Duyệt, lấy, sao chép, xóa file.....	14
II.4.4.6. Keylogger, khóa bàn phím	15
II.4.4.7. Chụp màn hình	17
II.4.4.8. Chụp ảnh, quay video từ webcam	17
III. Môi trường phát triển.....	18
IV. Hướng dẫn sử dụng.....	19
IV.1. Cách 1: Cài đặt file setup (dành cho users).....	19
IV.2. Cách 2: Build từ mã nguồn (dành cho developers).....	20
V. Đóng góp của các thành viên.....	21
VI. Tài liệu tham khảo	21

I. Giới thiệu

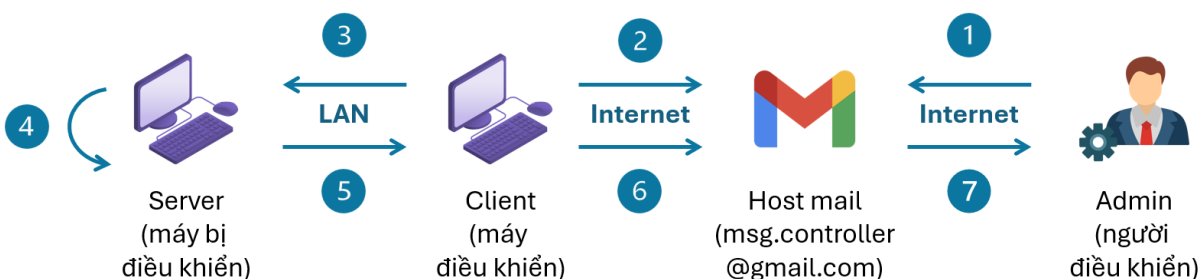
Ứng dụng **Remote PC Control via Email** được phát triển trong đồ án môn học Mạng máy tính, cho phép một người (admin) có thể điều khiển máy tính từ xa thông qua email. Chỉ cần biết địa chỉ IP của máy bị điều khiển (server), admin có thể gửi đến địa chỉ email **msg.controller@gmail.com** các yêu cầu như tắt/mở ứng dụng, chụp màn hình, theo dõi bàn phím, tắt máy,...

Trong quá trình thực hiện đồ án, nhóm không chỉ vận dụng những kiến thức đã học mà còn cố gắng tìm hiểu và áp dụng các kỹ thuật mới. Cụ thể, nhóm đã áp dụng **lập trình socket** để xây dựng giao tiếp hiệu quả giữa client và server, giúp truyền tải các yêu cầu và phản hồi một cách nhanh chóng. Bên cạnh đó, nhóm đã nghiên cứu và sử dụng **Gmail API** cùng với cơ chế xác thực OAuth2 để đảm bảo việc nhận và gửi email được thực hiện một cách an toàn và bảo mật. Còn về phía server, **Windows API** và thư viện **STL C++** được sử dụng để thực hiện quản lý và điều khiển tài nguyên hệ thống, từ đó hỗ trợ các yêu cầu điều khiển máy tính từ xa một cách hiệu quả.

Trong quá trình thử nghiệm, ứng dụng đã cơ bản đáp ứng được các yêu cầu đề ra, với khả năng thực hiện các chức năng chính như điều khiển và phản hồi qua email một cách hiệu quả. Tuy nhiên, ứng dụng không thể tránh khỏi những thiếu sót, đặc biệt khi xử lý các tình huống phức tạp. Do đó, nhóm rất mong nhận được ý kiến đóng góp của thầy để đồ án được hoàn thiện hơn.

II. Thiết kế đồ án

II.1. Luồng hoạt động của ứng dụng



Hình II.1: Luồng hoạt động của ứng dụng

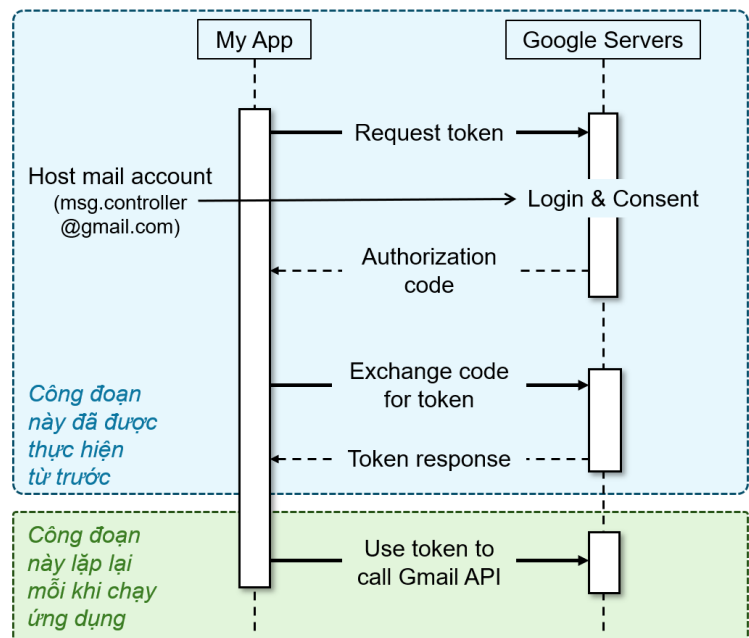
- (1) Admin gửi yêu cầu điều khiển server đến host mail, một email được xem là hợp lệ khi có:
- Subject (tiêu đề) bắt đầu bằng **[ctr1]** và theo sau là một lệnh điều khiển
 - Body (nội dung) là địa chỉ IP của server

- (2) Máy điều khiển (đã được cài đặt trình client) liên tục check mail từ hộp thư của host mail bằng cách sử dụng Gmail API. Khi có mail mới, client thực hiện:
 - Kiểm tra người gửi có quyền điều khiển hay không. Nếu không, thông báo cho người gửi.
 - Kiểm tra địa chỉ IP có hợp lệ hay không, Nếu không, thông báo cho người gửi.
- (3) Nếu người gửi có quyền điều khiển và địa chỉ IP hợp lệ, client sẽ mở kết nối TCP đến server bằng địa chỉ IP trên, sau đó gửi yêu cầu (subject của email trên) cho máy bị điều khiển (đã được cài đặt trình server) qua socket.
- (4) Sau khi nhận được yêu cầu của client qua socket, server sẽ kiểm tra tính hợp lệ của yêu cầu. Nếu hợp lệ, server sử dụng kết hợp Windows API và thư viện STL để thực hiện các yêu cầu.
- (5) Server gửi phản hồi cho client qua socket trên kết nối đã thiết lập trước đó.
- (6) Client sử dụng Gmail API để yêu cầu host mail gửi phản hồi trên cho admin.
- (7) Host mail gửi mail phản hồi cho admin, bao gồm: subject, body và file đính kèm (nếu có).

II.2. Host email

Trong hệ thống, `msg.controller@gmail.com` được sử dụng làm host email đóng vai trò trung tâm để giao tiếp giữa admin, client, và server. Địa chỉ email này được cấu hình để cho phép client thực hiện việc đọc các email đến và gửi phản hồi thông qua Gmail API.

Cụ thể, để đảm bảo tính an toàn và bảo mật, host email sẽ cấp quyền truy cập cụ thể cho client thông qua **cơ chế xác thực OAuth**. Client sử dụng một **access token** được cung cấp sau khi xác thực, cho phép thực hiện các tác vụ như đọc các email mới và gửi email phản hồi đến admin. Lưu ý rằng access token này sẽ hết hạn sau một khoảng thời gian nhất định, do đó ta phải làm mới (refresh) mỗi khi sử dụng để đảm bảo tính liên tục của hệ thống.



Hình II.2: `msg.controller@gmail.com` được cấu hình làm host email cho ứng dụng

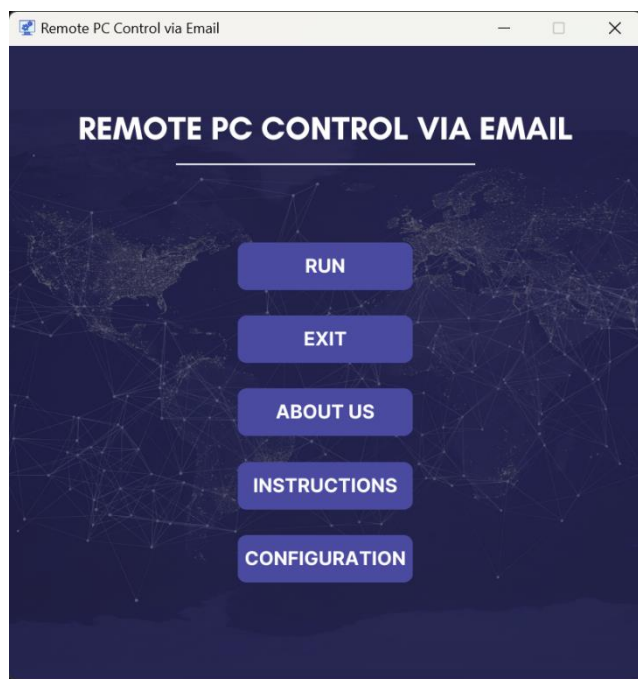
II.3. Client (máy điều khiển)

II.3.1. Giao diện app

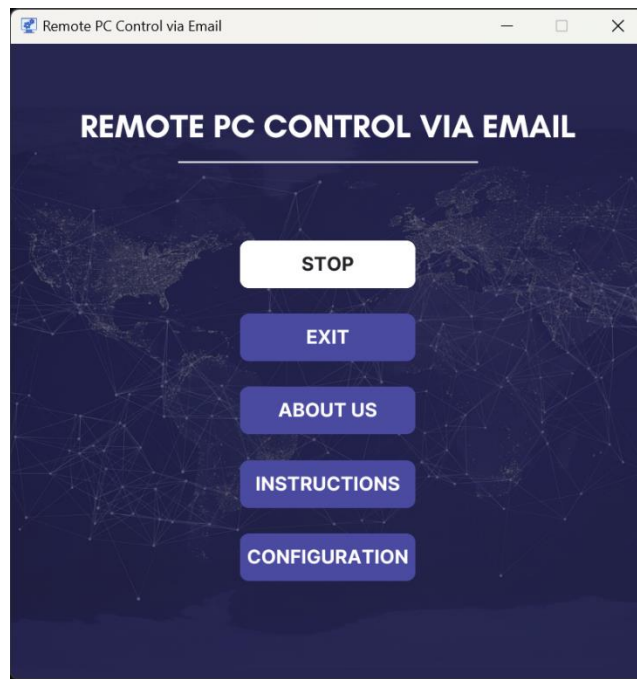
Ứng dụng client được thiết kế với giao diện đồ họa đơn giản, trực quan dựa trên **thư viện SFML**, giúp người dùng dễ dàng vận hành và quản lý hệ thống điều khiển máy tính từ xa.

Giao diện của ứng dụng cung cấp các chức năng chính như sau:

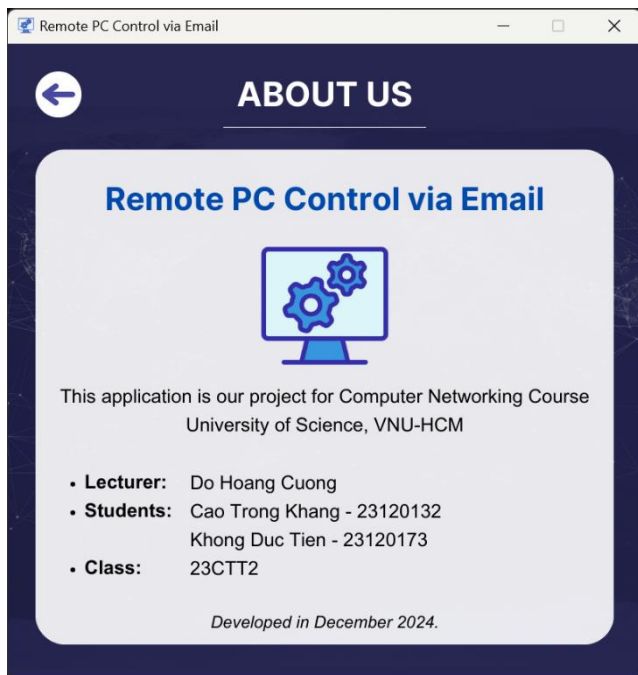
- **Run:** Bắt đầu kiểm tra các email mới liên tục từ hộp thư của host mail.
- **Stop:** Ngừng việc kiểm tra email.
- **Exit:** Thoát khỏi ứng dụng một cách an toàn.
- **About Us:** Hiện thị thông tin về nhóm phát triển ứng dụng.
- **Instructions:** Giới thiệu cách hoạt động của ứng dụng, giúp người dùng hiểu rõ hơn về cơ chế vận hành.
- **Configuration:** Cho phép cấu hình danh sách admin thông qua việc thêm và xóa (thông tin này sẽ được lưu trong file `admin.txt`), đảm bảo hệ thống chỉ nhận lệnh từ các địa chỉ email được cấp quyền.



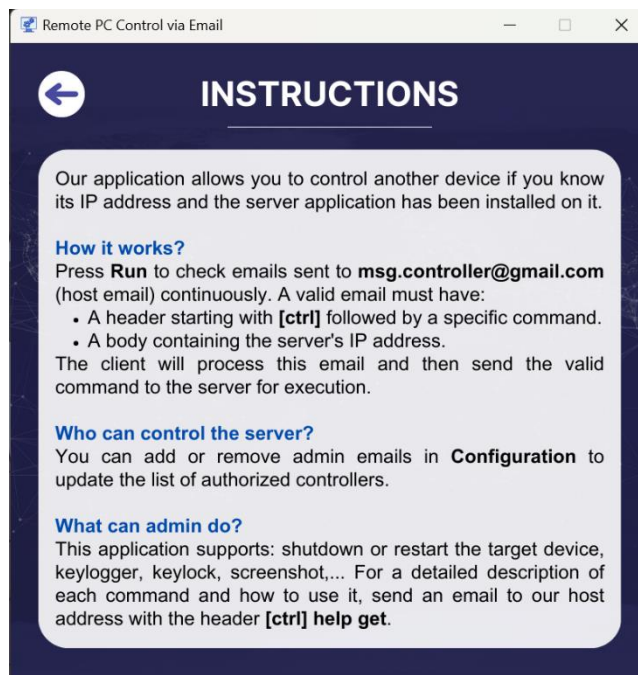
Hình II.3: Homepage trước khi nhấn nút Run



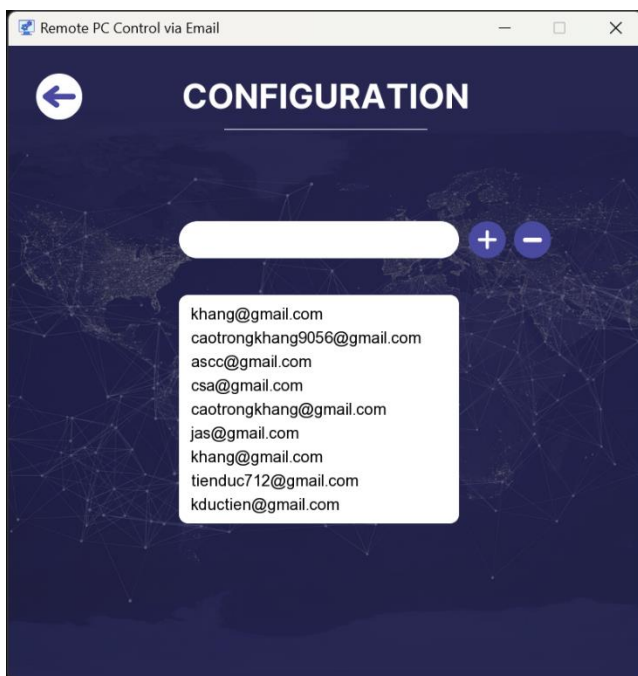
Hình II.4: Homepage sau khi nhấn nút Run



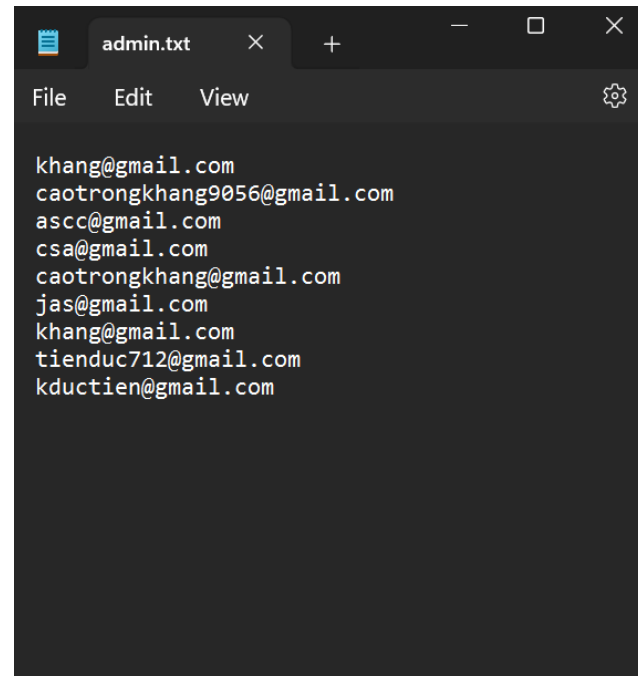
Hình II.5: Giao diện About Us



Hình II.6: Giao diện Instructions



Hình II.7: Giao diện Configuration



Hình II.8: File admin.txt lưu thông tin

II.3.2. Các chức năng chính

II.3.2.1. Kiểm tra mail và gửi mail phản hồi

a. Hàm checkMail()

Mã giả:

Kiểm tra mailbox của host email

Nếu có email mới

Đánh dấu email là đã đọc

Lấy subject, body và địa chỉ email của người gửi

Nếu người gửi không hợp lệ

Gửi mail phản hồi: Access denied

Nếu địa chỉ IP không hợp lệ

Gửi mail phản hồi: Invalid IP address

Nếu cả người gửi và địa chỉ IP đều hợp lệ

runClient()

Gửi mail phản hồi với thông tin nhận được từ server

b. Hàm sendMail()

- Cung cấp đầy đủ thông tin cần thiết: địa chỉ email người gửi, người nhận, tiêu đề, phần thân email và file đính kèm (nếu có).
- Mã hóa base64 cho toàn bộ email trước khi gửi.

II.3.2.2. Gửi yêu cầu của admin cho server và nhận phản hồi

a. Hàm runClient()

Mã giả:

Khởi tạo winsock: WSASStartup()

Tạo socket cho client, sử dụng giao thức TCP: socket()

Kết nối đến server: connect()

Gửi request đến server: send()

// Chờ server thực hiện yêu cầu...

Nhận phản hồi từ server: receiveData()

Đóng kết nối: closesocket(), WSACleanup()

b. Hàm receiveData()

- Mã giả:

Nhận kích thước của tiêu đề

Nhận tiêu đề

Nhận kích thước của nội dung email

Nhận nội dung email

Nhận kích thước của dữ liệu file

Nếu không có dữ liệu, kết thúc

Nếu dữ liệu quá lớn (giới hạn 50MB), kết thúc

Ngược lại

Nhận kích thước tên file

Nhận tên file

Tính kích thước của dữ liệu file

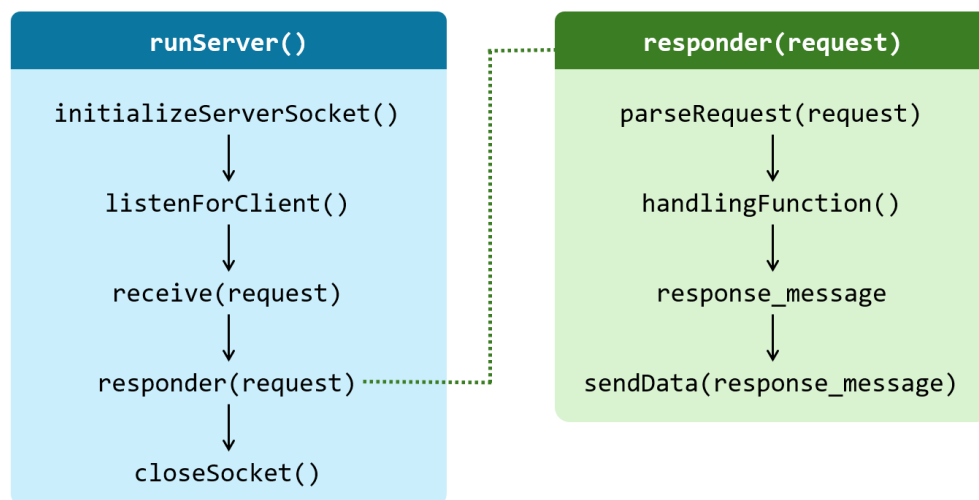
Nhận dữ liệu file

- Khi phản hồi từ server có kích thước lớn, client không thể nhận hết trong một lần do giới hạn về kích thước bộ đệm và băng thông mạng. Do đó, ta cần chia nhỏ dữ liệu thành nhiều phần và nhận lần lượt từng phần để đảm bảo không bị lỗi bộ đệm hoặc mất dữ liệu. Quá trình này bao gồm các bước:
 - Nhận kích thước của từng phần dữ liệu trước khi thực hiện nhận dữ liệu thực tế.
 - Tiến hành nhận từng phần nhỏ của dữ liệu.
 - Lặp lại quá trình cho đến khi nhận toàn bộ dữ liệu từ server.

II.4. Server (máy bị điều khiển)

II.4.1. Nhận yêu cầu và phản hồi cho client

- Tương tự như client, server cũng thực hiện khởi tạo socket và sử dụng giao thức TCP. Sau khi lắng nghe và thiết lập kết nối, server nhận yêu cầu từ client. Khi đó hàm `responder()` sẽ xử lý để lấy thông điệp phản hồi và gửi lại cho client. Kết nối đóng sau khi gửi xong.



Hình II.9: Server giao tiếp với client

- Đối với vấn đề kích thước của phản hồi là rất lớn đã được đề cập trong phần hàm `receiveData()` phía client, hàm `sendData()` của server cũng có cách xử lý tương tự khi chia nhỏ dữ liệu thành nhiều phần và gửi lần lượt từng phần để đảm bảo không bị lỗi bộ đệm hoặc mất dữ liệu.

II.4.2. Phân tích yêu cầu từ client

- Cấu trúc của lệnh điều khiển (là phần header trong email của admin) hợp lệ:

`[ctrl] <từ khóa 1> <từ khóa 2> <các tham số>`

Trong đó:

- `[ctrl]`: Tiền tố để nhận biết đây là một lệnh điều khiển
- `<từ khóa 1>`: Danh mục điều khiển chính (đối tượng như file, process, key,...)
- `<từ khóa 2>`: Hành động được thực hiện (động tác như get, delete, list...)
- `<các tham số>`: Danh sách tham số đi kèm (có thể là tên file, đường dẫn, thời gian...)
- Cây ánh xạ `requestTree` được xây dựng bằng cấu trúc dữ liệu map với mỗi nhánh gồm một ánh xạ từ lệnh điều khiển sang hàm xử lý tương ứng.
- Hàm `parseRequest()`

Mã giả:

Tách chuỗi lệnh thành danh sách các token

Xác minh token đầu tiên là tiền tố `[ctrl]`

Nếu sai, trả về "Wrong request format"

Tìm từ khóa 1 trong `requestTree`

Nếu không tìm thấy, trả về "Command not found"

Tìm từ khóa 2 trong `requestTree`

Nếu không tìm thấy, trả về "Command not found"

So sánh số token còn lại với số lượng tham số cần thiết

Nếu khác nhau, trả về "Insufficient arguments"

Trả về hàm xử lý (`outFunction`) và danh sách tham số (`outParams`)

Trả về thông báo "Parse request successfully"

- Sau đó, hàm `responder()` (mục II.4.1) sẽ gọi hàm xử lý để lấy thông điệp phản hồi.

II.4.3. Xây dựng thông điệp phản hồi

- Để đảm bảo tính đồng bộ và thuận tiện trong việc xử lý kết quả, tất cả các hàm xử lý lệnh điều khiển trong server đều trả về kiểu `Response`, được định nghĩa như sau:

`typedef pair<string, vector<BYTE>> Response`

Trong đó:

- **Thành phần thứ nhất** (string): Là nội dung phần thân email (body email), chứa kết quả của hàm xử lý lệnh điều khiển.
 - **Thành phần thứ hai** (vector<BYTE>): Là dữ liệu file, bao gồm tên file và nội dung được ghép lại thành một vector. Nếu file không được yêu cầu thì vector này rỗng.
- Phần thân email nói trên luôn được **định dạng HTML**, giúp thông điệp phản hồi được trình bày một cách trực quan, dễ đọc, đồng thời hỗ trợ hiển thị các thông tin phức tạp như bảng dữ liệu hoặc cấu trúc cây thư mục.

II.4.4. Xử lý các lệnh điều khiển

Phần này sẽ lần lượt trình bày từng lệnh điều khiển theo yêu cầu của đồ án, các hàm xử lý hỗ trợ bởi Windows API và hình ảnh mail phản hồi tương ứng.

II.4.4.1. Liệt kê các lệnh điều khiển

Lệnh: [ctrl] help get

This mail responses to the request: [ctrl] help get

Note:

- All commands must start with [ctrl] and be written in lowercase.
- <duration> must be a positive integer.
- <path> must be enclosed in double quotes (") if it contains spaces, and directories must be separated by a slash (/). Additionally, all characters in the path must be Latin alphabet characters. For example: <path> = "D:/HCMUS.1/HK3/MMT/Project/Description/Day la mot vi du.jpg".

Type	Command	Description
HELP	help get	Show the command list of this application.
APP	app list	Get the list of running apps on the device.
	app stop <ID>	Close the running app with the given ID.
	app start <path>	Start the app at the specified path.
PROCESS	process list	Get the list of running processes on the device.
	process stop <ID>	Close the running process with the given ID.
	process start <path>	Start the process at the specified path.

Hình II.10: Phản hồi lệnh **help get**

II.4.4.2. Liệt kê, tắt, mở tiến trình (process)

- Liệt kê tiến trình

- Lệnh: `[ctrl] process list`
- Xử lý: Dùng hàm `CreateToolhelp32Snapshot()` để lấy danh sách các tiến trình đang chạy trên máy, sau đó lấy thông tin của từng tiến trình (tên, ID, số luồng đang chạy) và đưa vào bảng.

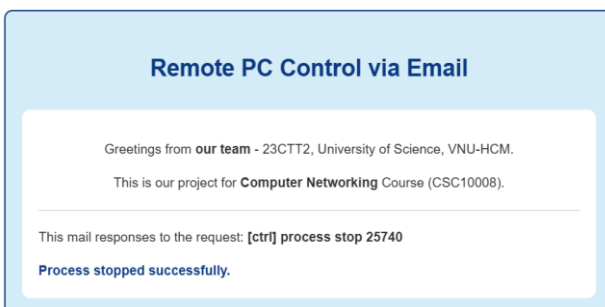
This mail responses to the request: **[ctrl] process list**

No.	Process	ID	Thread Count
1	[System Process]	0	1
2	System	4	1
3	Secure System	184	1
4	Registry	252	1
5	smss.exe	760	1
6	csrss.exe	1072	2

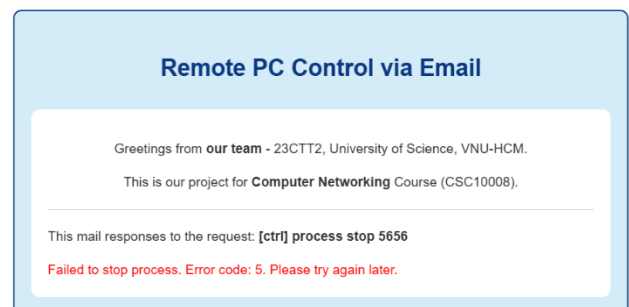
Hình II.11: Phản hồi lệnh **process list**

- Tắt tiến trình

- Lệnh: `[ctrl] process stop <ID>` (ID được lấy từ bảng liệt kê các tiến trình)
- Xử lý: Dùng hàm `CloseHandle()`. Lưu ý rằng một số tiến trình được Windows bảo vệ hoặc đang chạy với quyền cao (Administrator hoặc System) không thể bị tắt bởi hàm xử lý thông thường.



Hình II.12: Phản hồi **process stop** (thành công)

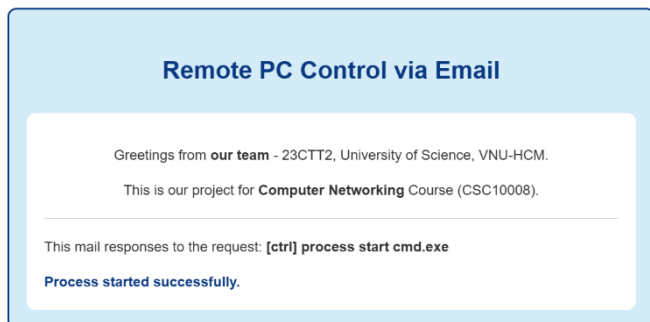


Hình II.13: Phản hồi **process stop** (thất bại)

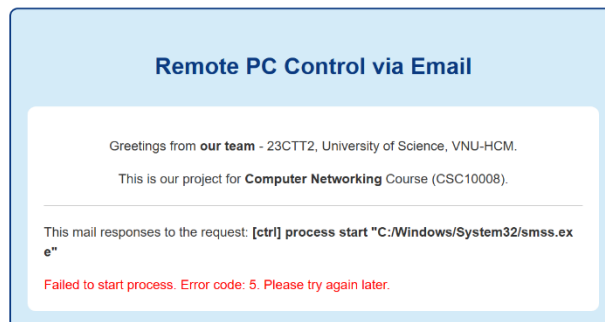
- Mở tiến trình

- Lệnh: `[ctrl] process start <path>` (path là đường dẫn đến tiến trình: .exe, .bat)

- Xử lý: Dùng hàm `ShellExecuteA()`. Lưu ý rằng một số tiến trình không thể mở bằng hàm xử lý thông thường do thiếu quyền truy cập, tệp không tồn tại, bị khóa, hoặc bị chặn bởi phần mềm bảo mật.



Hình II.14: Phản hồi **process start** (thành công)



Hình II.15: Phản hồi **process start** (thất bại)

II.4.4.3. Liệt kê, tắt, mở ứng dụng (application)

- Liệt kê ứng dụng

- Lệnh: `[ctrl] app list`
- Xử lý: Tương tự như liệt kê tiến trình. Tuy nhiên, ta phải kiểm tra và chỉ hiển thị những tiến trình nào có giao diện người dùng.

This mail responses to the request: **[ctrl] app list**

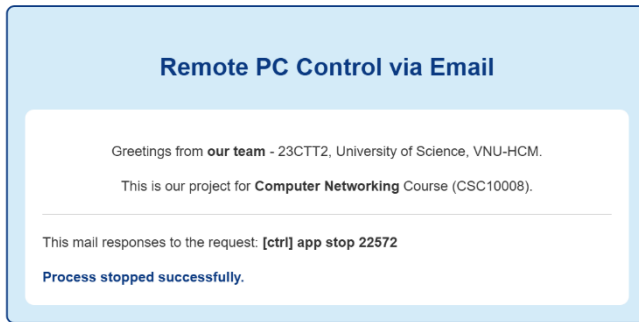
No.	Application	ID
1	chrome.exe	5732
2	ApplicationFrameHost.exe	13556
3	devenv.exe	5888
4	VsDebugConsole.exe	21136
5	SystemSettings.exe	27108
6	explorer.exe	9292
7	TextInputHost.exe	10728
8	WINWORD.EXE	11372

Hình II.16: Phản hồi lệnh **app list**

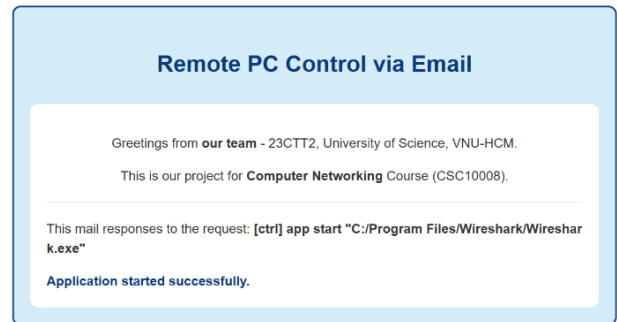
- Tắt ứng dụng

- Lệnh: `[ctrl] app stop <ID>` (ID được lấy từ bảng liệt kê các ứng dụng)
- Xử lý: Hoàn toàn tương tự với tắt tiến trình.

- Mở ứng dụng
 - Lệnh: [ctrl] app start <path> (path là đường dẫn đến ứng dụng: .exe)
 - Xử lý: Hoàn toàn tương tự với mở tiến trình. Ngoài ra, có một số ứng dụng thường được thêm vào biến môi trường PATH của hệ thống nên chỉ cần nhập tên file mà không cần đường dẫn đầy đủ: notepad.exe, cmd.exe, calc.exe, powershell.exe, mspaint.exe, taskmgr.exe, explorer.exe, control.exe (Hình II.14).



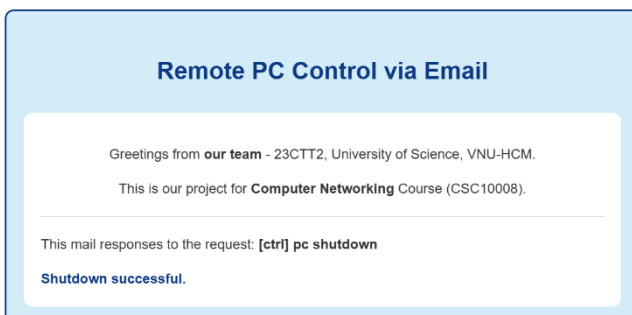
Hình II.17: Phản hồi lệnh **app stop**



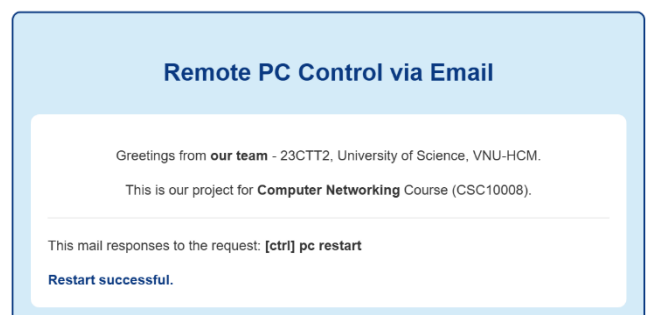
Hình II.18: Phản hồi lệnh **app start**

II.4.4.4. Shutdown, Restart

- Shutdown
 - Lệnh: [ctrl] pc shutdown
 - Xử lý: Đầu tiên, chương trình mở token của tiến trình hiện tại để điều chỉnh đặc quyền, sau đó kích hoạt đặc quyền tắt máy (SE_SHUTDOWN_NAME). Tiếp theo, API InitiateSystemShutdownEx được gọi để gửi lệnh shutdown với các tham số như thời gian chờ, lý do tắt máy, và tùy chọn ép đóng ứng dụng.
- Restart
 - Lệnh: [ctrl] pc restart
 - Xử lý: Tương tự như khi xử lý lệnh shutdown, tuy nhiên lúc này API InitiateSystemShutdownEx được gọi để gửi lệnh restart.



Hình II.19: Phản hồi lệnh **pc shutdown**



Hình II.20: Phản hồi lệnh **pc restart**

II.4.4.5. Duyệt, lấy, sao chép, xóa file

- Duyệt file

- Lệnh: [ctrl] file explore <path> (path là đường dẫn đến thư mục cần duyệt)
- Xử lý: Nếu path = “”, trả về danh sách các ổ đĩa trên máy tính. Nếu path là một folder, duyệt qua tất cả các mục trong folder và trả về danh sách các mục đó. Cuối cùng, danh sách trên sẽ được hàm html_tree() xử lý để vẽ cây thư mục.

This mail responses to the request: [ctrl] file explore "D:/HCMUS.1/HK3/MMT/Project/Remote Control/ServerApp"

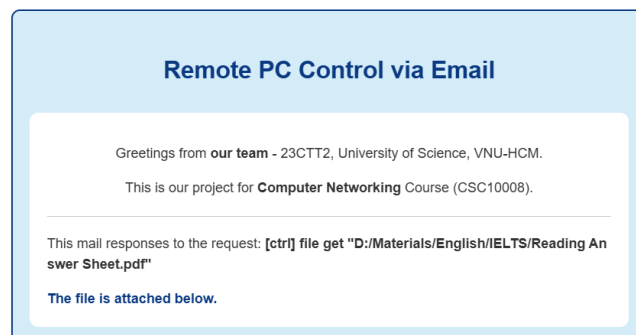
The directory tree for D:/HCMUS.1/HK3/MMT/Project/Remote Control/ServerApp/ (1 - level):

```
D:/HCMUS.1/HK3/MMT/Project/Remote Control/ServerApp/  
├─ html_generator.cpp  
├─ include/  
├─ request_handle.cpp  
├─ ServerApp.vcxproj  
├─ ServerApp.vcxproj.filters  
├─ ServerApp.vcxproj.user  
├─ service/  
├─ smain.cpp  
├─ ssocket.cpp  
└─ x64/
```

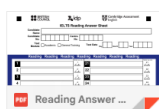
Hình II.21: Phản hồi lệnh *file explore*

- Lấy file

- Lệnh: [ctrl] file get <path> (path là đường dẫn đến file cần lấy)
- Xử lý: Dựa trên đường dẫn được cung cấp, chương trình sẽ đọc file và lưu dữ liệu (bao gồm cả tên file) vào thành phần thứ hai của response (mục II.4.3).

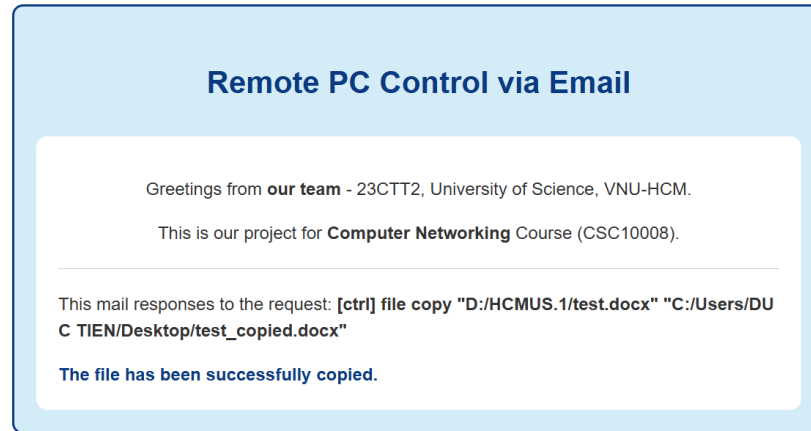


Một tệp đính kèm • Gmail đã quét



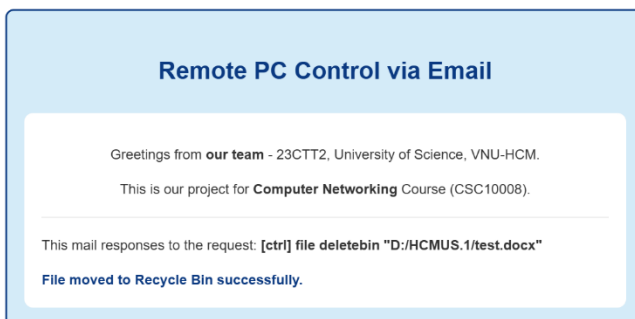
Hình II.22: Phản hồi lệnh *file get*

- Sao chép file
 - Lệnh: `[ctrl] file copy <source> <dest>` (source là đường dẫn đến file nguồn, dest là đường dẫn đến file đích)
 - Xử lý: Dùng hàm `CopyFileA()`.

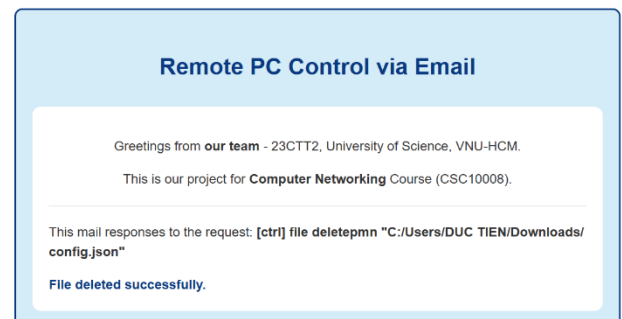


Hình II.23: Phản hồi lệnh **file copy**

- Xóa file tạm thời (di chuyển file vào Recycle Bin)
 - Lệnh: `[ctrl] file deletebin <path>` (path là đường dẫn đến file cần xóa)
 - Xử lý: Dùng hàm `SHFileOperationW()`.
- Xóa file vĩnh viễn
 - Lệnh: `[ctrl] file deletepmn <path>` (path là đường dẫn đến file cần xóa)
 - Xử lý: Dùng hàm `DeleteFileA()`.



Hình II.24: Phản hồi lệnh **file deletebin**

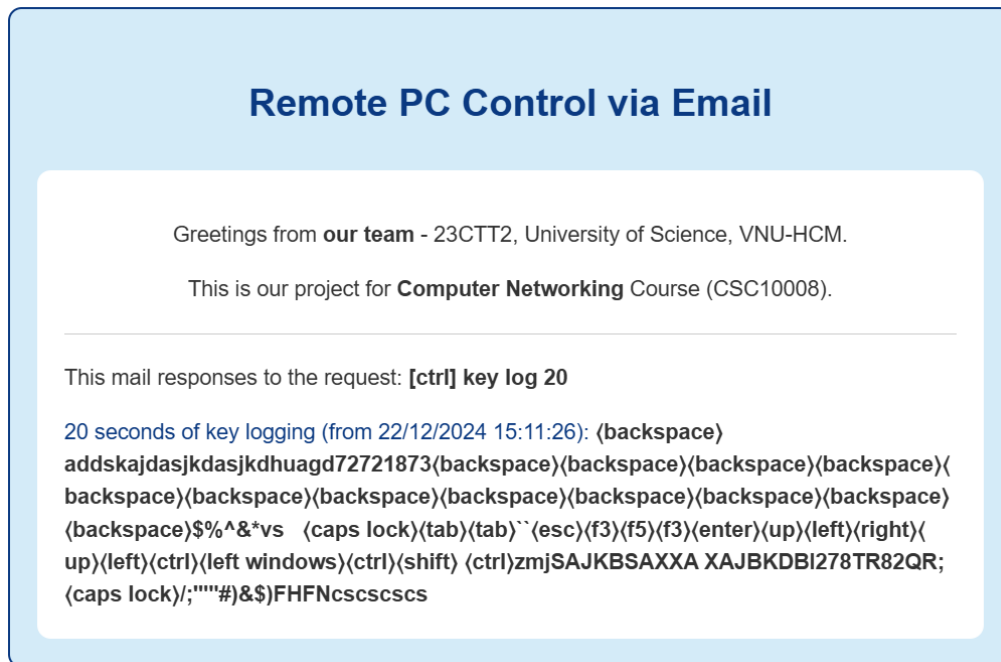


Hình II.25: Phản hồi lệnh **file deletepmn**

II.4.4.6. Keylogger, khóa bàn phím

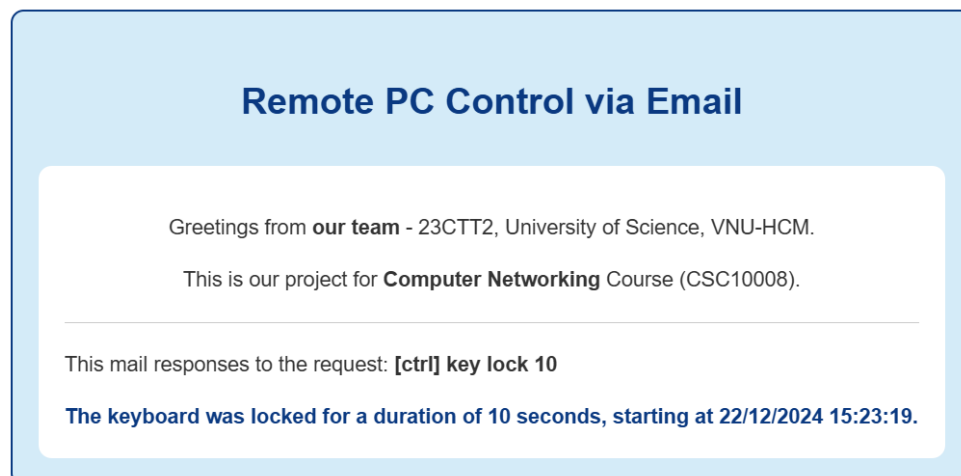
- Keylogger
 - Lệnh: `[ctrl] key log <duration>` (duration là thời gian thực hiện keylogger)
 - Xử lý: Dùng hàm `GetAsyncKeyState()` để ghi nhận các phím được gõ từ bàn phím, đồng thời cần phải xử lý trường hợp khi người dùng nhấn tổ hợp phím hoặc các

phím đặc biệt. Trong thông điệp phản hồi, các phím đặc biệt sẽ được đặt trong cặp ký tự < > (không có trên bàn phím) để dễ phân biệt với các phím thông thường.



Hình II.26: Phản hồi lệnh **key log**

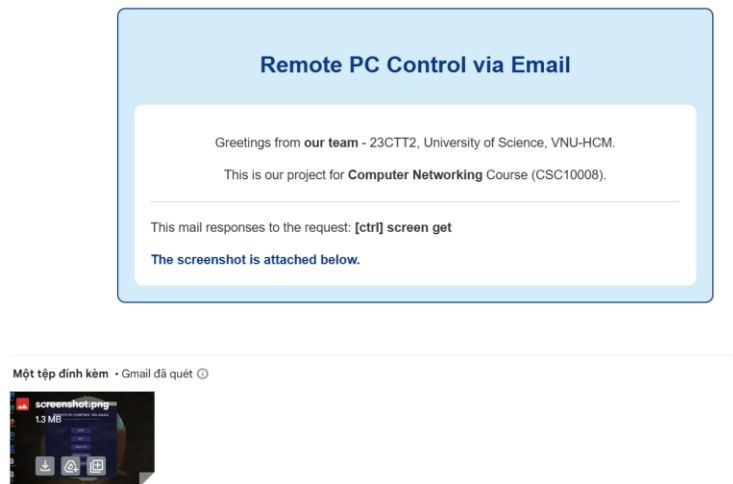
- Khóa bàn phím
 - Lệnh: `[ctrl] key lock <duration>` (duration là thời gian khóa bàn phím)
 - Xử lý: Dùng một **keyboard hook** (WH_KEYBOARD_LL) để chặn các sự kiện phím. Một luồng riêng được tạo ra để thiết lập và duy trì hook, trong đó hàm `LowLevelKeyboardProc` kiểm tra các sự kiện bàn phím. Nếu chế độ khóa được đang kích hoạt (`isKeyboardLocked`), mọi phím sẽ bị chặn. Sau khoảng thời gian được định trước, chế độ khóa sẽ tự động tắt, hook được gỡ bỏ và luồng xử lý kết thúc.



Hình II.27: Phản hồi lệnh **key lock**

II.4.4.7. Chụp màn hình

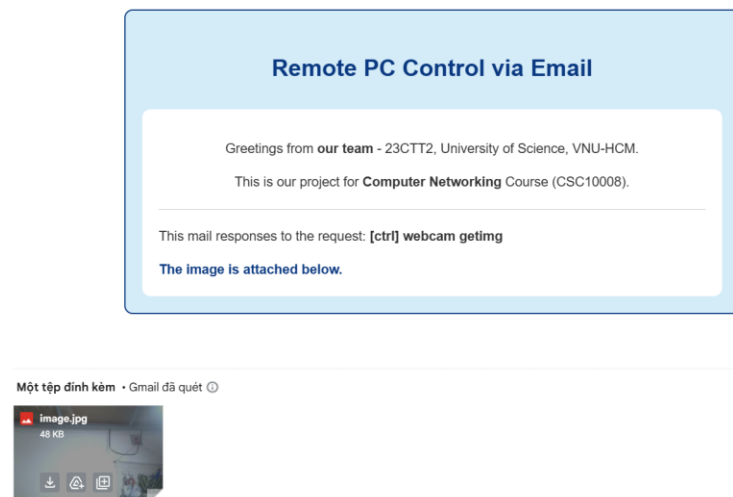
- Lệnh: `[ctrl] screen get`
- Xử lý: Đầu tiên kích thước màn hình được lấy bởi hàm `GetSystemMetricsForDpi()`, sau đó hàm `BitBlt()` thực hiện sao chép nội dung màn hình vào một bitmap. Dữ liệu từ bitmap đó được chuyển vào đối tượng `cv::Mat` và ghi ra file bằng lệnh `cv::imwrite`. Cuối cùng chỉ cần đọc file vào thông điệp phản hồi.



Hình II.28: Phản hồi lệnh *screen get*

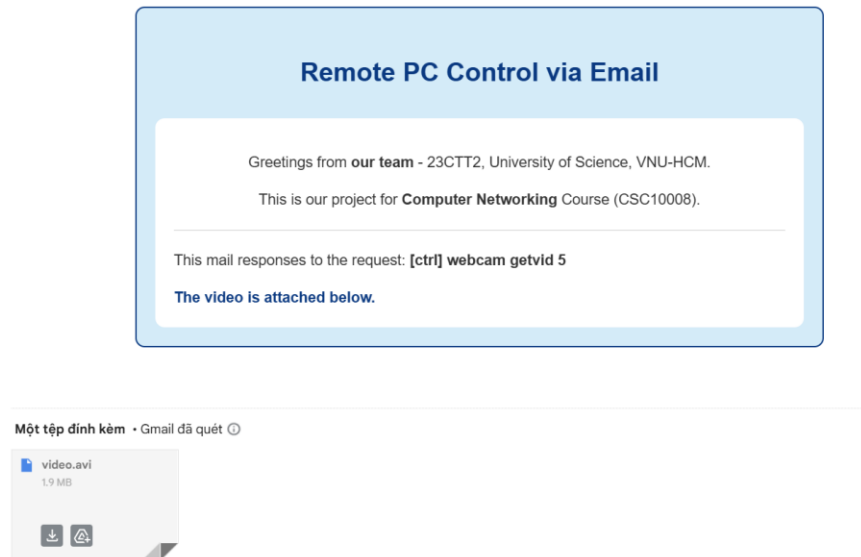
II.4.4.8. Chụp ảnh, quay video từ webcam

- Chụp ảnh từ webcam
 - Lệnh: `[ctrl] webcam getimg`
 - Xử lý: Sau khi webcam được mở bằng `cv::VideoCapture` và chờ 500ms để ổn định, một khung hình được lấy từ webcam và lưu vào đối tượng `cv::Mat`. Đối tượng này sau đó được xử lý tương tự ở phần chụp màn hình (mục II.4.4.7).



Hình II.29: Phản hồi lệnh *webcam getimg*

- Quay video từ webcam
 - Lệnh: `[ctrl] webcam getvid <duration>` (duration là độ dài thời gian của video)
 - Xử lý: Sau khi webcam được mở bằng `cv::VideoCapture`, một đối tượng `cv::VideoWriter` được tạo để ghi video với tên file và định dạng cụ thể. Video được quay với tốc độ khung hình (fps) xác định và số lượng khung hình tương ứng với thời gian quay (duration). Các khung hình liên tục được lấy từ webcam và ghi vào tệp video.



Hình II.30: Phản hồi lệnh **webcam getvid**

III. Môi trường phát triển

- Hệ điều hành: Windows 11 64-bit
- Ngôn ngữ lập trình: C++ (tiêu chuẩn C++ 17)
- IDE: Visual Studio 2022
- Framework hỗ trợ giao diện: SFML
- Thư viện sử dụng:
 - STL (Standard Template Library): Cung cấp các cấu trúc dữ liệu và thuật toán cơ bản
 - Winsock2: Hỗ trợ lập trình socket trên Windows
 - OpenCV: Xử lý và phân tích hình ảnh
 - cURL: Gửi yêu cầu HTTP/HTTPS, làm việc với các API web
 - OpenSSL: Cung cấp các tính năng mã hóa và bảo mật
 - Windows API: Giao tiếp với các tính năng cấp thấp của hệ điều hành Windows
 - Gmail API: Kết nối và xử lý email từ tài khoản Gmail

IV. Hướng dẫn sử dụng

IV.1. Cách 1: Cài đặt file setup (dành cho users)

- Truy cập thư mục của Google Drive [tại đây](#). Trong đó, người dùng sẽ tải file **RemoteControl-Server.msi** nếu muốn chạy chương trình server và tải file **RemoteControl-Client.msi** nếu muốn chạy chương trình client.
- Cài đặt chương trình server:
 - + **Bước 1:** Mở file .msi vừa tải về. Nếu máy hiện thông báo như *Hình IV.1*, chọn **More info** rồi chọn tiếp **Run anyway** như *Hình IV.2*.

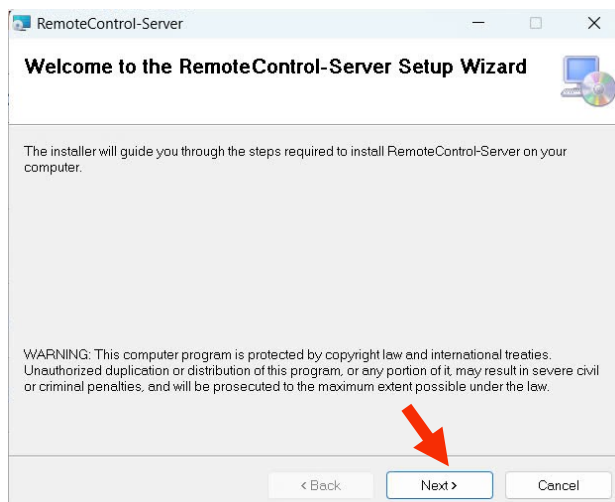


Hình IV.1: Windows cảnh báo người dùng

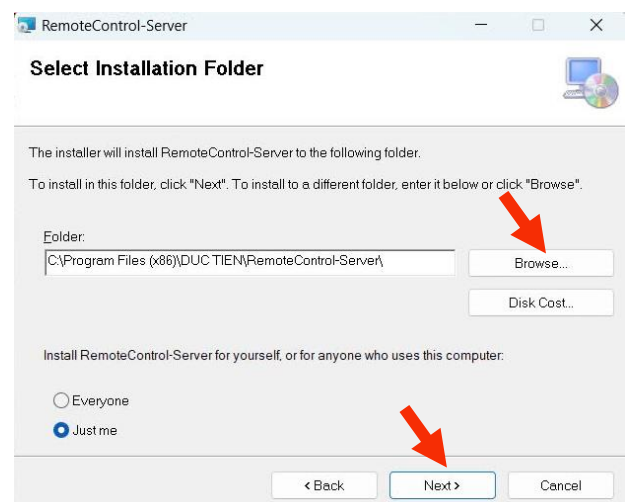


Hình IV.2: Nhấn Run anyway để tiếp tục

- + **Bước 2:** Ở màn hình Welcome, chọn **Next** (*Hình IV.3*). Sau đó, ở màn hình Select Installation Folder, người dùng tự chọn thư mục để chứa chương trình (bao gồm file .exe và các thư viện cần thiết) rồi chọn **Next** để tiếp tục (*Hình IV.4*).

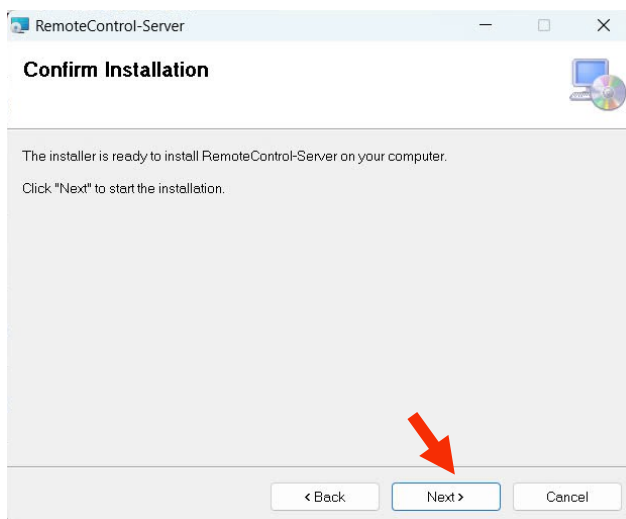


Hình IV.3: Chọn Next để tiếp tục

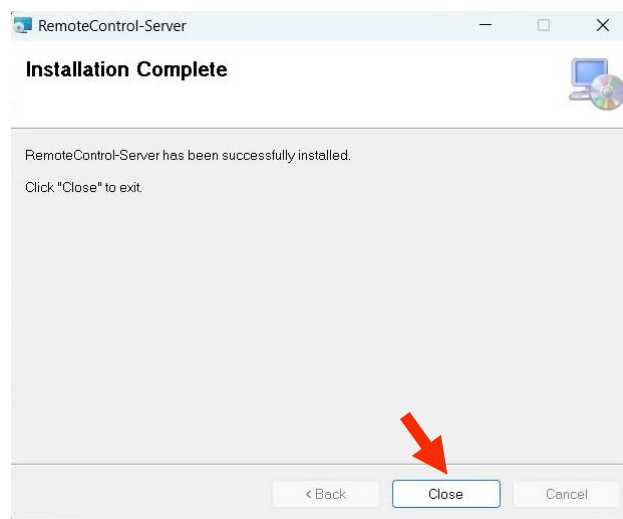


Hình IV.4: Chọn thư mục chứa chương trình

- + **Bước 3:** Ở màn hình Confirm Installation, chọn **Next** để bắt đầu quá trình cài đặt (Hình IV.5). Sau khoảng vài giây, màn hình Installation Complete hiện lên thông báo đã cài đặt thành công, chọn **Close** để kết thúc (Hình IV.6).



Hình IV.5: Xác nhận cài đặt



Hình IV.6: Cài đặt hoàn tất

- + **Bước 4:** Ở màn hình Desktop sẽ xuất hiện shortcut có tên là **RC-Server**. Người dùng mở file này để chạy chương trình server.
- Cài đặt chương trình client: Hoàn toàn tương tự với chương trình server. Sau khi cài đặt hoàn tất, ở màn hình Desktop sẽ xuất hiện shortcut có tên là **RC-Client**. Người dùng mở file này để chạy chương trình client.

IV.2. Cách 2: Build từ mã nguồn (dành cho developers)

Phần này đã được trình bày trong phần đầu của video demo đồ án. Có thể xem [tại đây](#).

V. Đóng góp của các thành viên

Bảng phân công bên dưới chỉ mang tính chất tương đối vì trong quá trình thực hiện, có những phần công việc có sự giao thoa nhau và các công việc được điều chỉnh linh hoạt giữa các thành viên để phù hợp với thực tế.

Thành viên	Công việc	Mức độ hoàn thành
Cao Trọng Khang 23120132	Thiết kế giao diện app client	100%
	Đọc mail mới, gửi mail phản hồi	100%
	Chức năng liệt kê, tắt, mở tiến trình	95%
	Chức năng liệt kê, tắt, mở ứng dụng	95%
	Chức năng shutdown, restart	100%
Khổng Đức Tiến 23120173	Thiết kế định dạng HTML cho mail phản hồi	100%
	Giao tiếp giữa client và server	100%
	Chức năng duyệt, lấy, sao chép, xóa file	100%
	Chức năng keylogger, khóa bàn phím	95%
	Chức năng chụp ảnh màn hình	100%
	Chức năng chụp ảnh, quay video từ webcam	100%

VI. Tài liệu tham khảo

- [1] Document hướng dẫn sử dụng Winsock API:
<https://learn.microsoft.com/en-us/windows/win32/winsock/>
- [2] Xác thực OAuth2 để dùng Gmail API:
<https://developers.google.com/identity/protocols/oauth2>
- [3] Mã hóa base64: <https://github.com/ReneNyffenegger/cpp-base64>
- [4] Các chức năng dùng Windows API:
<https://learn.microsoft.com/en-us/windows/win32/apiindex/windows-api-list>
- [5] Chức năng chụp ảnh màn hình:
<https://learn.microsoft.com/en-us/windows/win32/gdi/capturing-an-image>
- [6] Định dạng HTML cho mail phản hồi:
<https://github.com/htrvu/remote-control-with-email-service>