Đại học Quốc gia Thành phố Hồ Chí Minh

Trường Đại học Khoa học Tự nhiên

ĐỒ ÁN
SỬ DỤNG EMAIL HỖ TRỢ ĐIỀU KHIỂN QUẢN TRỊ MÁY TÍNH TỪ XA

Nhóm sinh viên thực hiện:

20120012 - Nguyễn Phạm Nhật Huy

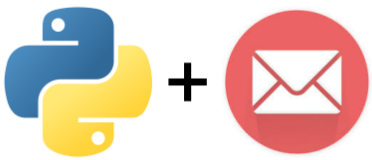
20120014 - Vương Gia Huy

20120021 - Hồ Văn Sơn

Giảng viên hướng dẫn: Đỗ Hoàng Cường

Lớp 20TN

Học phần: Mạng máy tính



06 - 2022

Mục Lục

1 Tìm hiểu sơ lược 2
1.1 Gửi email thông qua Python 2
1.2 Truy xuất email thông qua python 2
1.3 Điều khiển máy tính với Python 2

2 Chương trình sử dụng email hỗ trợ điều khiển quản trị máy tính từ xa . . 3
2.1 Cách chương trình hoạt động 3
2.1.1 Client 3
2.1.2 Host Server 3
2.2 Chi tiết cài đặt 3
2.2.1 Listener 3
2.2.2 Các chức năng 4
2.2.3 Giao diện 6

3 Tiêu chí đánh giá 11
3.1 Tự đánh giá 11
3.2 Bảng phân công công việc 11

4 Tài liệu tham khảo 12

1. Tìm hiểu sơ lược

Trong những năm gần đây, cùng với sự phát triển mạnh mẽ của ngành viễn thông, nhu cầu sử dụng máy tính cá nhân một cách gián tiếp mọi lúc, mọi nơi là vô cùng thiết yếu. Bắt kịp xu hướng đó, nhóm chúng tôi đã triển khai một ý tưởng là sử dụng email để điều khiển quản trị máy tính từ xa được viết bằng ngôn ngữ lập trình Python.

1.1. Gửi email thông qua Python

Python được tích hợp sẵn một thư viện có tên là `smtplib`. Thư viện này cho phép bạn tạo ra một client **SMTP** và có thể dùng client này để gửi thư đến bất kì thiết bị nào có kết nối Internet.

SMTP (Simple Mail Transfer Protocol) là một giao thức ở tầng Application (giao thức tầng Transport là TCP) được dùng để truyền tải thư điện tử (e-mail) trên mạng Internet. Nó thiết lập kênh kết nối giữa mail client và mail server và thiết lập kênh liên lạc giữa mail server gửi và mail server nhận. Email sẽ được truyền từ mail client lên mail server và được mail server này gửi đi đến mail server nhận.

Có thể tham khảo thêm tại đây: SMTP - Lý thuyết, SMTP - Triển khai.

1.2. Truy xuất email thông qua python

Python cũng được tích hợp một thư viện khác có tên là `imaplib`. Thư viện này cho phép bạn truy cập email thông qua giao thức **IMAP**.

IMAP (Internet Message Access Protocol) là một giao thức truyền mail giúp ta có thể truy cập tài khoản và đọc email từ bất kỳ thiết bị nào. Khi đọc email bằng IMAP, dữ liệu không thực sự tải xuống hoặc lưu trữ trên máy tính. Do đó, ta có thể kiểm tra email trong cả điều kiện băng thông thấp.

Có thể tham khảo thêm tại đây: IMAP - Lý thuyết, IMAP - Triển khai.

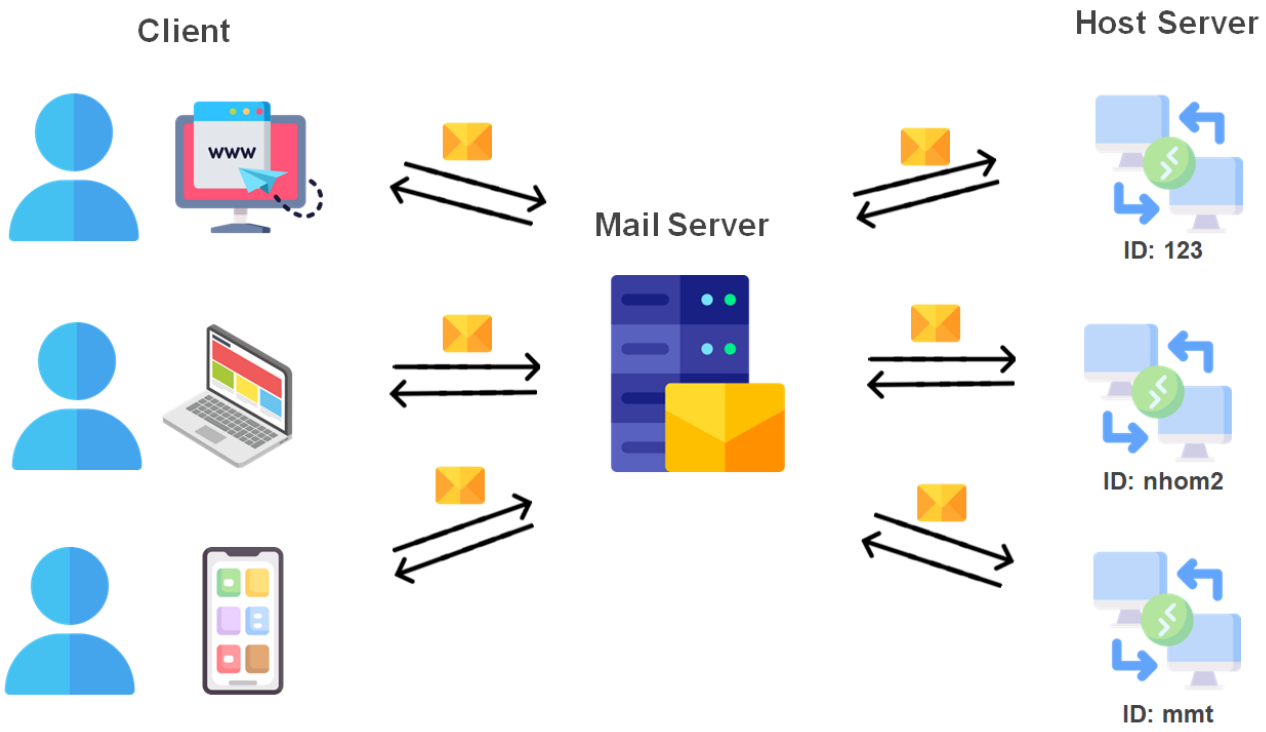
1.3. Điều khiển máy tính với Python

Python là một ngôn ngữ lập trình mạnh mẽ, hiện đại. Nhờ sự phong phú của các thư viện bổ sung, Python cho phép bạn viết các đoạn lệnh để điều khiển phần cứng máy tính như webcam, microphone, chuột, bàn phím, thiết bị mạng, ... một cách dễ dàng và nhanh chóng.

Có thể tìm hiểu thêm tại đây: Control PC with Python.

2. Chương trình sử dụng email hỗ trợ điều khiển quản trị máy tính từ xa

2.1. Cách chương trình hoạt động



2.1.1. Client

Để điều khiển một máy tính, ta cần tạo một email và gửi đến mail server (địa chỉ mail server mà nhóm sử dụng là `mmt.nhom2@gmail.com`) có subject là `'ID: <ID host server trên máy tính muốn điều khiển>'` và nội dung là lệnh điều khiển tương ứng. Sau khi email được gửi đến mail server, các host server đang hoạt động sẽ tự động tìm kiếm những email có ID của host server đó, thực hiện xử lý lệnh điều khiển và phản hồi lại kết quả người dùng.

2.1.2. Host Server

Host server sẽ nhận lệnh điều khiển từ mail server thông qua Internet Message Access Protocol (IMAP), xử lí và phản hồi thông qua Simple Mail Transfer Protocol (SMTP).

Mỗi host server được định danh bằng một ID duy nhất được lưu trữ trên mail server. Chương trình sẽ tự sinh ngẫu nhiên một ID gồm 6 chữ số cho host server. Ngoài ra, người dùng cũng có thể tuỳ chỉnh thành một ID bất kì dễ nhớ hơn miễn là ID đó chưa được lưu trữ trên mail server.

2.2. Chi tiết cài đặt

2.2.1. Listener

Là lớp đối tượng đại diện cho máy tính (host server) ta cần điều khiển từ xa. Lớp này sử dụng thư viện `imaplib` để truy xuất email, xử lý và sử dụng thư viện `smtplib` để gửi email phản hồi. Các phương thức của lớp này là:

- __init__(self, SMTP_HOST, IMAP_HOST, SERVER_ADDRESS, SERVER_PASSWORD, screenInfo): là phương thức khởi tạo được truyền vào các tham số STMP_HOST là địa chỉ SMTP server, IMAP_HOST là địa chỉ IMAP server, SERVER_ADDRESS là địa chỉ mail server, SERVER_PASSWORD là mật khẩu để đăng nhập mail server, screenInfo là thông tin kích thước màn hình máy tính để hỗ trợ một số chức năng. Nhóm đã sử dụng Gmail SMTP server `smtp.gmail.com`, Gmail IMAP server `imap.gmail.com`, mail server do nhóm tự tạo `mmt.nhom2@gmail.com`.
- __del__(self): là phương thức hủy, để tắt server.
- login(self, ADDR, PASS): là phương thức đăng nhập với tham số truyền vào là địa chỉ mail server và mật khẩu để đăng nhập.
- create_ID(self, ID): là phương thức kiểm tra và tạo ID cho máy tính đang chạy chương trình với tham số truyền vào là ID cần sử dụng.
- delete_ID(self, delID): là phương thức xóa ID của máy tính đang chạy chương trình với tham số truyền vào là ID đang được sử dụng.
- download_file(self, msg, path): là phương thức cho phép tải tệp về đường dẫn path.
- get_request(self, msg): là phương thức nhận lệnh điều khiển từ mail server.
- handle_msg(self, msg): là phương thức xử lí lệnh điều khiển.
- send_msg(self, subject, content): là phương thức gửi phản hồi với tham số truyền vào là chủ đề và nội dung cần phàn hồi.
- recv_msg(self): là phương thức lắng nghe thông tin từ mail server.

2.2.2. Các chức năng

DIRECTORY

- DIRECTORY LIST_TREE <path>: in ra cây thư mục gốc là thư mục <path>. Lệnh điều khiển này được xử lý bởi hàm listTree(path).
- DIRECTORY LIST_DIR <path>: in ra tất cả thư mục, tập tin nằm trong thư mục <path>. Lệnh điều khiển này được xử lý bởi hàm listDir(path).
- DIRECTORY LIST_DISK: in ra tất cả ổ đĩa trong máy tính. Lệnh điều khiển này được xử lý bởi hàm listDisk().
- DIRECTORY DELETE <path>: xóa thư mục hoặc tập tin có đường dẫn <path>. Lệnh điều khiển này được xử lý bởi hàm deleteDF(path).
- DIRECTORY COPY <src> <dst>: sao chép tập tin hoặc thư mục từ đường dẫn <src> đến đường dẫn <dst>. Lệnh điều khiển này được xử lý bởi hàm copy(src, dst).
- DIRECTORY SAVE_FILE <path>: lưu các tập tin đính kèm từ email vào thư mục <path>. Lệnh điều khiển này được xử lý bởi phương thức download_file(msg, path) của lớp Listener.

PROCESS

- PROCESS LIST_APPS: in ra tất cả ứng dụng đang chạy kèm port ID và thread. Lệnh điều khiển này được xử lý bởi hàm list_apps().
- PROCESS LIST_PROCESSES: in ra tất cả tiến trình đang chạy kèm port ID và thread. Lệnh điều khiển này được xử lý bởi hàm list_processes().
- PROCESS KILL <PortID>: tắt tiến trình đang chạy tại port <PortID>. Lệnh điều khiển này được xử lý bởi hàm kill(pid).
- PROCESS START <App>: mở ứng dụng <App>. Lệnh điều khiển này được xử lý bởi hàm start(name).

KEYLOGGER

- KEYLOGGER HOOK: tắt/mở theo dõi bàn phím. Lệnh điều khiển này được xử lý bởi hàm keylogger(key), sẽ có 2 trường hợp xảy ra được đánh dấu bởi cờ ishook:
  - ishook = 0 có nghĩa là bàn phím không bị theo dõi. Lúc này, host server sẽ thực hiện hành động mở theo dõi, giá trị ishook sẽ được gán bằng 1.
  - ishook = 1 có nghĩa là bàn phím đang bị theo dõi. Lúc này, host server sẽ thực hiện hành động tắt theo dõi, giá trị ishook sẽ được gán bằng 0.
- KEYLOGGER PRINT: in ra các phím đã nhấn và mốc thời gian tương ứng. Lệnh điều khiển này được xử lý bởi hàm listKey2HTML(ls).
- KEYLOGGER LOCK: khóa bàn phím. Lệnh điều khiển này được xử lý bởi hàm lock().
- KEYLOGGER UNLOCK: mở khóa bàn phím. Lệnh điều khiển này được xử lý bởi hàm unlock().

SCREEN

- SCREEN TAKE: chụp màn hình. Lệnh điều khiển này được xử lý bởi hàm take_screen().
- SCREEN CAPTURE <n_seconds>: ghi màn hình <n_seconds> giây. Lệnh điều khiển này được xử lý bởi hàm capture_screen(n_seconds, dim).

CAMERA

- CAMERA <n_seconds>: quay camera <n_seconds> giây. Lệnh điều khiển này được xử lý bởi hàm record_camera(n_seconds).

REGISTRY

- REGISTRY GET_VALUE <Path> <name_value>: lấy giá trị của <name_value> tại key có đường dẫn <Path> của registry. Lệnh điều khiển này được xử lý bởi hàm get_value(full_path).
- REGISTRY SET_VALUE <Path> <name_value> <value> <type>: đặt giá trị của <name_value> tại key có đường dẫn <Path> của registry thành giá trị <value> với kiểu dữ liệu là <type> (<type> là một trong những giá trị ["REG_SZ", "REG_BINARY", "REG_DWORD", "REG_QWORD", "REG_MULTI_SZ", "REG_EXPAND_SZ"]). Lệnh điều khiển này được xử lý bởi hàm set_value(full_path, value, value_type).

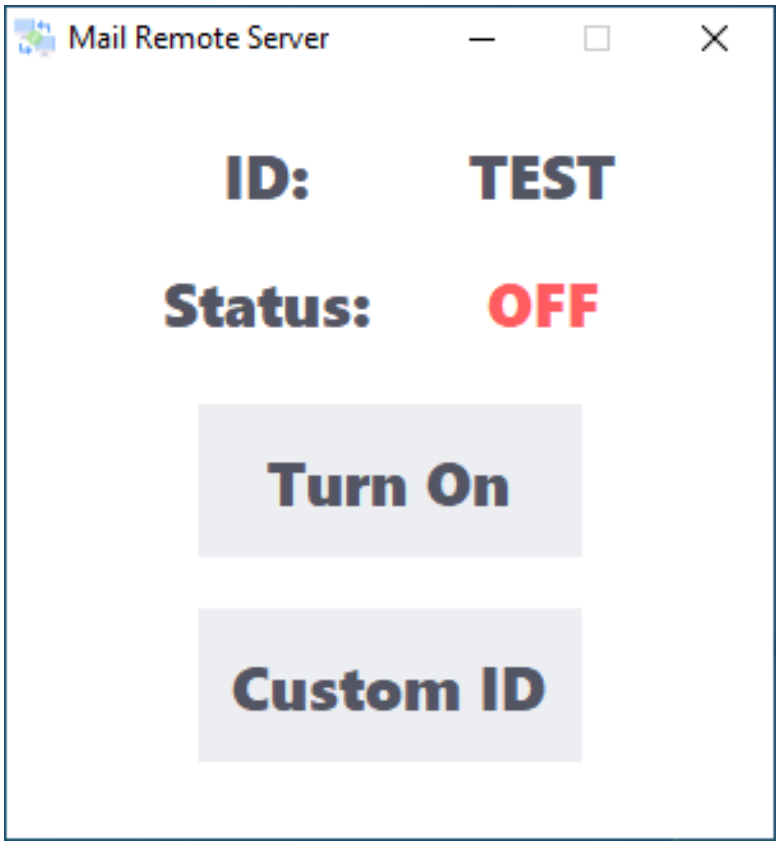
- REGISTRY CREATE_KEY <Path>: tạo key tại đường dẫn <Path>. Lệnh điều khiển này được xử lý bởi hàm create_key(full_path).
- REGISTRY DELETE_KEY <Path>: xóa key tại đường dẫn <Path>. Lệnh điều khiển này được xử lý bởi hàm delete_key(full_path).

SHUTDOWN & RESTART & LOGOUT

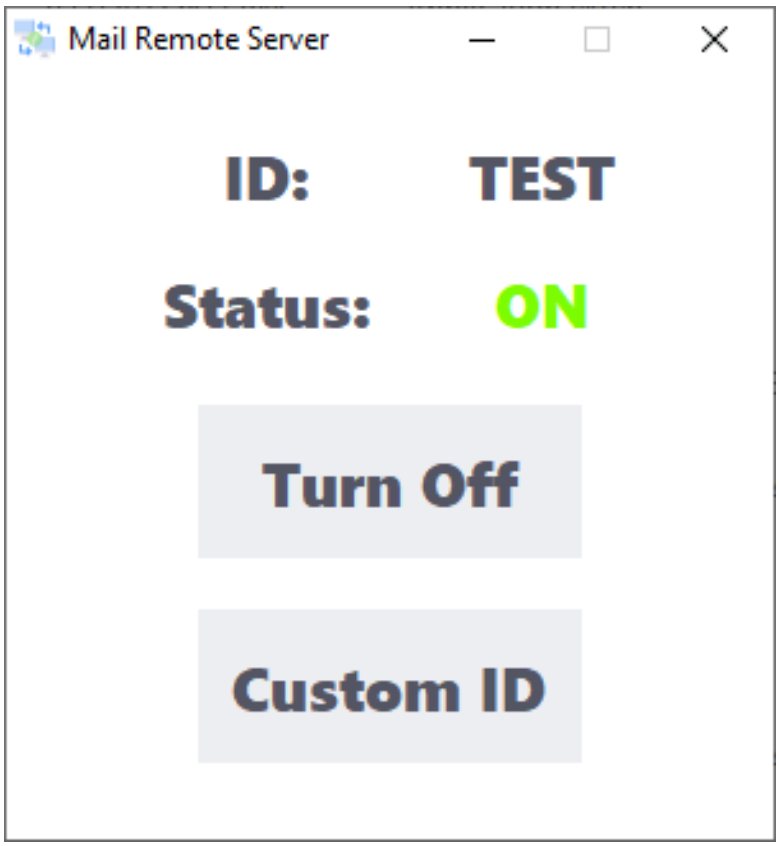
- SHUTDOWN: tắt máy. Lệnh điều khiển này được xử lý bởi hàm shutdown().
- RESTART: khởi động lại máy. Lệnh điều khiển này được xử lý bởi hàm restart().
- LOGOUT: đăng xuất tài khoản hiện tại trên máy. Lệnh điều khiển này được xử lý bởi hàm logout().

2.2.3. Giao diện

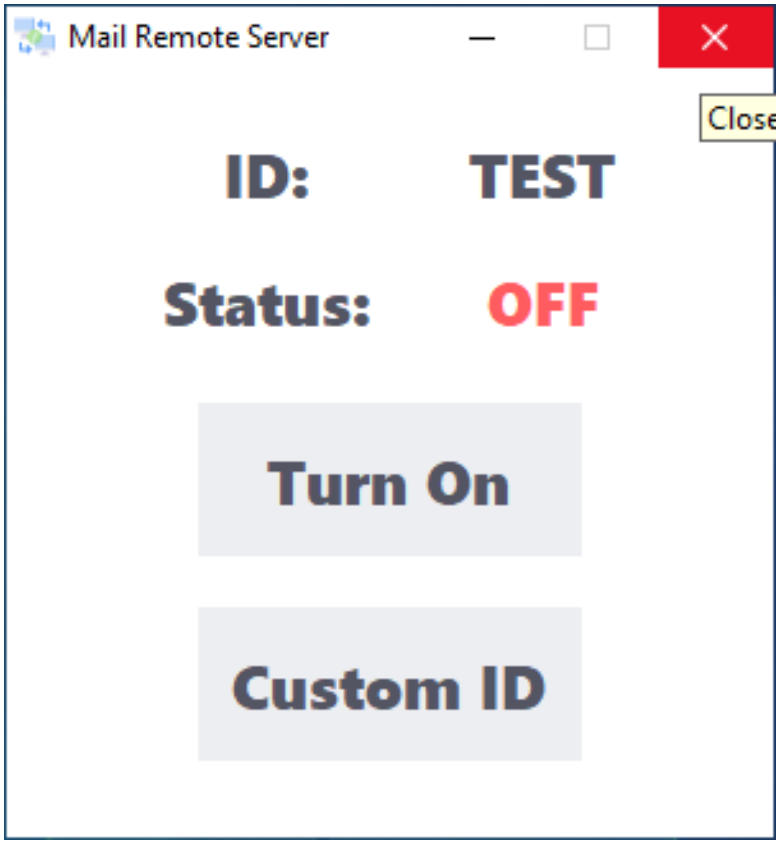
Trạng thái bình thường



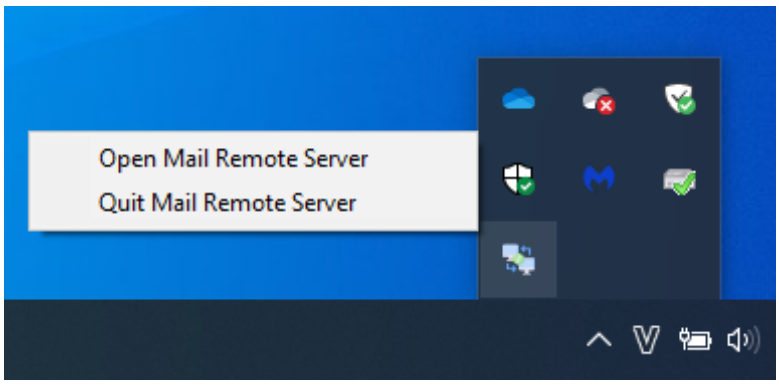
Đây là giao diện khi chạy chương trình ở trạng thái bình thường. Nhấn vào nút Turn On để bắt đầu lắng nghe thông tin từ mail server, khi đó Status sẽ chuyển sang trạng thái **On** và nút Turn On sẽ chuyển thành nút Turn Off.



Ngược lại, để dừng lắng nghe thông tin từ mail server, ta nhấn vào nút Turn Off, khi đó Status sẽ trở về trạng thái **Off** và nút Turn Off sẽ chuyển thành nút Turn On.



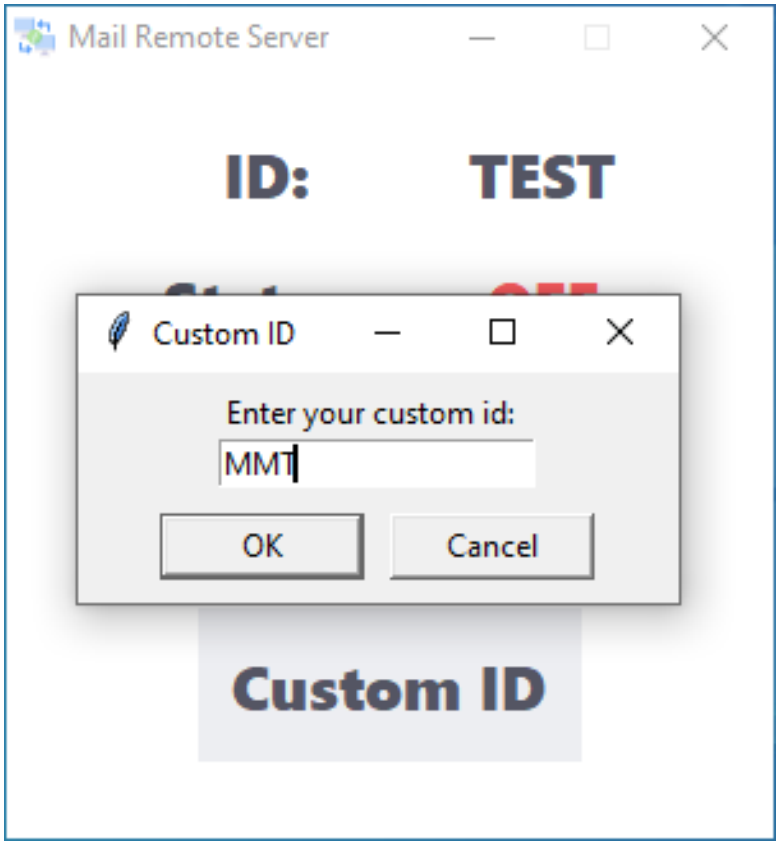
Ta có thể chuyển chương trình sang trạng thái chạy ngầm bằng cách nhấn vào biểu tượng **Close**.



Chương trình sẽ thu nhỏ trở thành một icon trong khay hệ thống và tiếp tục lắng nghe từ mail server nếu đang ở trạng thái hoạt động.

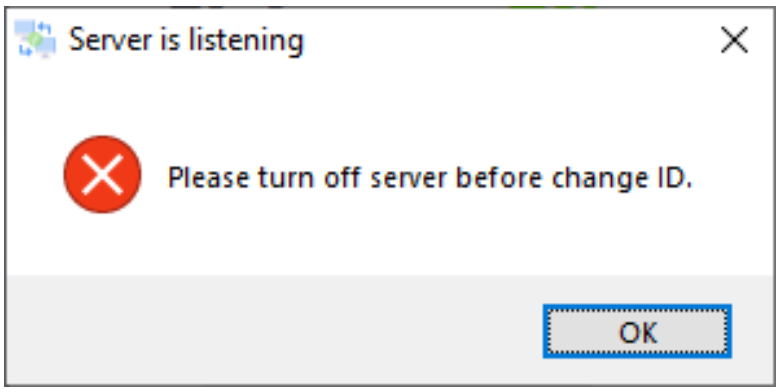
Ở trạng thái này, chương trình có hai lựa chọn:

- Open Mail Remote Server: mở lại giao diện bình thường của chương trình.
- Quit Mail Remote Server: thoát chương trình.

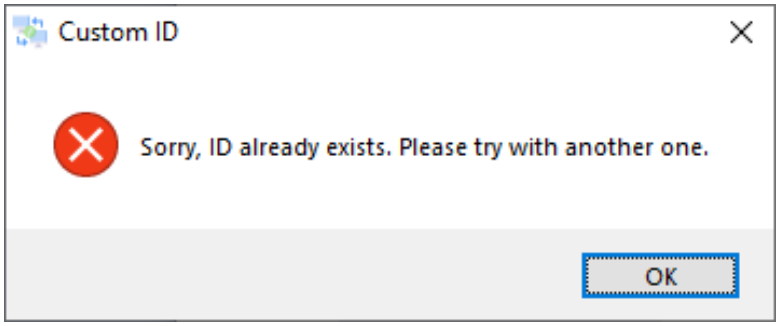


Khi ta muốn tuỳ chỉnh ID, ta có thể nhấn vào nút Custom ID và điền ID mới vào hộp thoại hiện lên.

Một số thông báo lỗi có thể gặp:

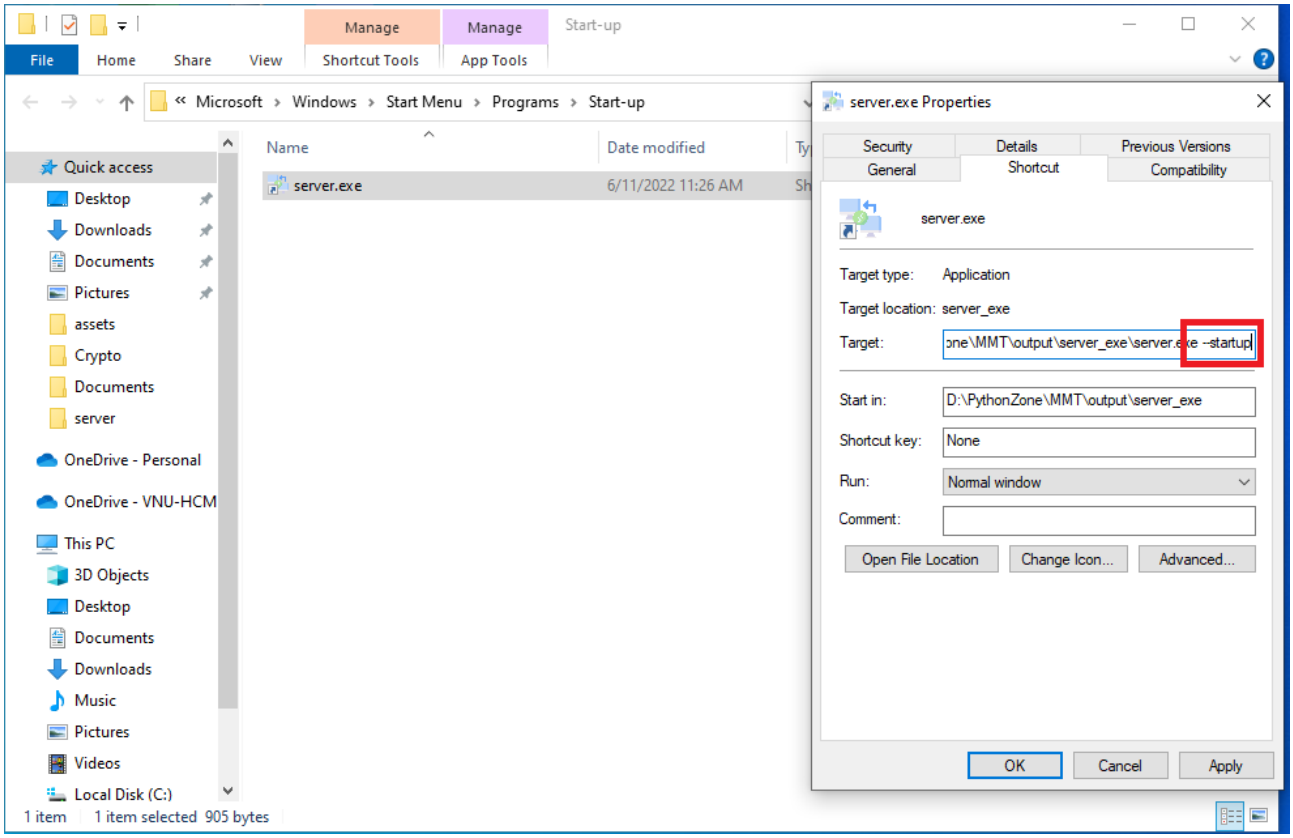


Lỗi này có nghĩa là host server đang ở trạng thái **On** và đang lắng nghe từ mail server. Ta cần dừng việc lắng nghe lại trước khi thay đổi ID.

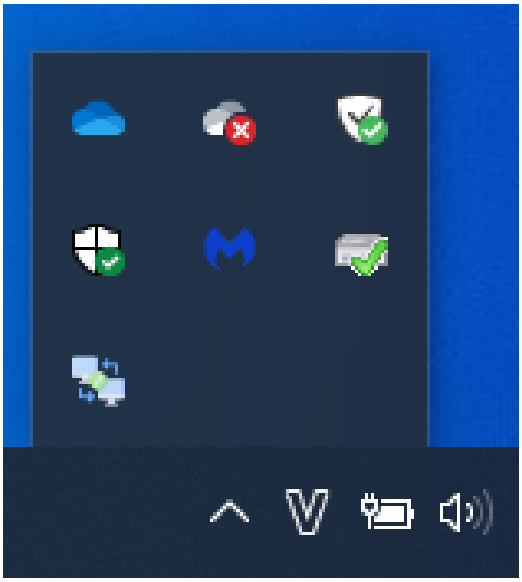


Lỗi này có nghĩa là ID mà ta muốn đổi đã có máy tính khác sử dụng. Ta phải sử dụng một ID khác.

Trạng thái tự khởi chạy



Ta có thể tạo một shortcut của chương trình trong thư mục Start-up của Windows và thêm tham số `--startup` để chương trình tự khởi chạy mỗi khi máy tính được bật.



Khi khởi động máy tính, chương trình sẽ tự khởi chạy và hiển thị ở trạng thái chạy ngầm và lắng nghe thông tin từ server.

3. Tiêu chí đánh giá

3.1. Tự đánh giá

- Thái độ làm việc của từng thành viên trong nhóm chuyên nghiệp, không trì trệ, có thái độ nghiệm túc, tích cực tìm hiểu về các câu hỏi của đồ án.
- Phối hợp công việc tốt, ăn ý.
- Phân công công việc công bằng, rõ ràng, rành mạch.

3.2. Bảng phân công công việc

MSSV	Họ và tên	Nội dung công việc	Mức độ hoàn thành
20120012	Nguyễn Phạm Nhật Huy	Code giao diện và tối ưu một số chức năng	100%
20120014	Vương Gia Huy	Đánh báo cáo latex, kiểm tra và đóng góp ý kiến về các chức năng	100%
20120021	Hồ Văn Sơn	Code về các hàm chức năng điều khiển máy tính	100%

4. Tài liệu tham khảo

Các nguồn tài liệu tham khảo:

- *Giáo trình mạng máy tính - ĐH KHTN TPHCM*. NXB Khoa học và Kỹ thuật, 2016
- *Tài liệu lý thuyết, thực hành do giảng viên cung cấp trên Moodle môn học*