

Trường đại học Khoa học Tự nhiên
Khoa Công nghệ Thông tin



BÁO CÁO ĐỒ ÁN ĐIỀU KHIỂN
MÁY TÍNH QUA EMAIL

Môn: Mạng máy tính

Nhóm sinh viên thực hiện:

20120018 – Trần Kiều Minh Lâm

20120067 – Nguyễn Phạm Bá Duy

20120316 – Nguyễn Hoàng Lâm

Giảng viên hướng dẫn: Đỗ Hoàng Cường

Năm học: 2021 – 2022

BẢNG PHÂN CÔNG THÀNH VIÊN

Tên thành viên	Tên công việc	Tự đánh giá
Trần Kiều Minh Lâm	Xây dựng backend, đọc mail từ mail server, release app, quay video	10/10
Nguyễn Phạm Bá Duy	Làm các yêu cầu điều khiển, GUI, viết báo cáo	10/10
Nguyễn Hoàng Lâm	Làm các yêu cầu điều khiển , gửi mail tới mail server, viết báo cáo	10/10

MỤC LỤC

Nội dung

I. Phân tích đề bài.....	4
II. Cấu trúc của email.	4
III. Cấu trúc các tập tin mã nguồn.	5
IV. Cách sử dụng chương trình.....	8
V. Giải thích một số thuật ngữ.....	10
VI. Nguồn tài liệu tham khảo	10

I. Phân tích đề bài

Đề bài: Tạo ứng dụng điều khiển máy tính qua việc gửi các email. Các yêu cầu bao gồm:

- Chụp màn hình
- Quay một đoạn video
- List các process
- Stop hoặc kill một process.
- Shut down
- Restart
- Copy file
- Bắt phím nhấn

Phân tích:

- Đầu tiên, ta cần tạo hàm thực hiện các yêu cầu nêu trên. Các hàm thực hiện yêu cầu sẽ nằm trong file [controller.py](#).
- Kế tiếp, ta cần kết nối ứng dụng với mail server để có thể đọc email và thực thi các yêu cầu đính kèm theo đó. Để kết nối ta dùng IMAP (Internet Message Access Protocol).
- Sau khi nhận email và thực hiện các yêu cầu, ta cần gửi mail trả lời cho những yêu cầu đó. Để có thể cho ứng dụng gửi mail ta dùng SMTP (Simple Mail Transfer Protocol).

Từ các phân tích trên ta tiến hành làm ứng dụng theo đó.

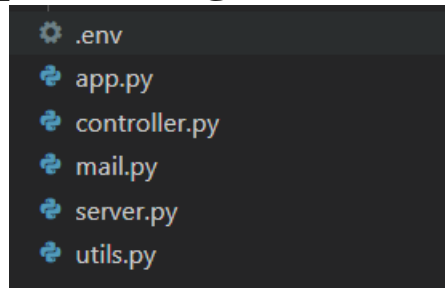
II. Cấu trúc của email.

Email có cấu trúc như sau:

- Envelope:
Gồm các thông tin cần thiết để gửi mail như địa chỉ người nhận, mức độ ưu tiên và bảo mật. Envelope dùng để định tuyến khi gửi mail.
- Header:
Gồm các thông tin mô tả mail đó. Gồm nhiều dòng, mỗi dòng là một cặp tên và giá trị (ngăn cách bởi dấu hai chấm). Các header cơ bản như: “To” lưu địa chỉ email người nhận, “Cc” lưu các địa chỉ người nhận khác, “From” lưu tên người gửi, “Sender” lưu địa chỉ mail người gửi, “Subject” lưu tiêu đề của mail.

- **Body:**
gồm các nội dung thực sự mà người dùng muốn gửi như text (chữ), html hoặc attachment (các file đính kèm).

III. Cấu trúc các tập tin mã nguồn.



Mã nguồn của chương trình bao gồm:

- **.env**: File lưu trữ các tham số cần thiết cho việc chạy chương trình như mail address, username, password, KEY, server address, port.
- **controller.py**: file lưu trữ các hàm thực hiện các yêu cầu từ người dùng.
 - Hàm **SHUTDOWN** và **RESTART**: gửi 1 email báo cáo kết quả thành công rồi sẽ shutdown máy. Sử dụng câu lệnh
`os.system('shutdown /s')`
có trong thư viện os.
 - Hàm **LIST_PROCESS**: mở terminal và chạy lệnh “tasklist”. Sau đó lưu lại danh sách các task rồi gửi email phản hồi. Sử dụng **popen()**:
`process_list = os.popen('tasklist').read()`
 - Hàm **STOP_PROCESS_BY_NAME** và **STOP_PROCESS_BY_ID** sẽ tách tên hoặc id lấy được từ yêu cầu, sau đó mở terminal và dùng lệnh “taskkill /f /im”.
 - Hàm **SCREENSHOT** sẽ chụp màn hình máy tính hiện tại. Sử dụng.
`ImageGrab.grab().save(file_path)`
trong thư viện PIL của python.

- Hàm **RECORD_VIDEO**: phân tách các tham số duration. Sau đó thực hiện record video thông qua hàm **record()** trong file **utils.py**. Hàm **record()** sẽ quay lại video bằng cách tổng hợp các frame liên tục, đúc kết thành file video, sau đó gửi video phản hồi cho người dùng.
- Hàm **COPY_FILE**: thực hiện copy một file thông qua terminal. Đầu tiên thực hiện phân tách tham số đầu vào, đích đến, sau đó dùng “popen” như đã trình bày ở trên.
- Hàm **RUN_COMMAND**: thực hiện 1 câu lệnh bất kì trên terminal. Kết quả trả về là kết quả trên terminal.
- Hàm **CATCH_KEYPRESS**: dùng để đọc các phím được nhấn trong một khoảng thời gian. Sử dụng module keyboard.
- Hàm **GET_REGISTRY**: dùng để lấy giá trị của một registry trong một registry path thông qua hàm **get_registry()** trong file **utils.py** sử dụng thư viện *winreg*
- Hàm **SET_REGISTRY**: tương tự hàm get registry, thông qua hàm **set_registry()** trong file **utils.py** để gán giá trị cho một registry trong một registry path.
- **mail.py**: file lưu các hàm thực hiện việc nhận mail, phân tách yêu cầu và gửi mail trả lời.
 - Hàm **fetch_mail** sử dụng thư viện **imaplib** trong python để tạo có thể login bằng tài khoản và mật khẩu vào mail server. Ở đây sử dụng mail server của Zoho với địa chỉ là **imap.zoho.com** và port là 993. Sau đó, đọc mail trong INBOX của tài khoản đó. Chỉ đọc những mail có key mà người dùng tạo xuất hiện trong tiêu đề của mail. Sau đó, ngắt kết nối là trả về danh sách các mail.
 - Hàm **create_mail** dùng để tạo một object MIMEMultipart trong thư viện email của python với các tham số gồm người nhận, chủ đề, nội dung, file đính kèm.

- Hàm `send_mail` sử dụng thư viện `smtplib` trong python để gửi mail đến người nhận. Bằng cách đăng nhập vào mail server của Zoho bằng địa chỉ `smtp.zoho.com` và port 465. Sau đó gửi mail object được tạo từ hàm `create_mail` và gửi tới người nhận. Cuối cùng sẽ ngắt kết nối tới mail server của Zoho.
- Lí do chọn mail server là Zoho: Ban đầu nhóm sử dụng Gmail nhưng sau đó chính sách bảo mật không cho phép các truy cập không an toàn (IMAP bị chặn) nên nhóm quyết định chuyển qua dùng Zoho.
- `server.py`: dùng để sử dụng các hàm của file `controller.py` và `mail.py` để tạo ra một server có thể đọc mail, xử lý mail theo yêu cầu và gửi mail phản hồi.
 - Tạo vòng lặp vô tận, gọi hàm `fetch_mail` trong `mail.py` và đọc mail liên tục. Khi có mail mới, gọi các hàm xử lý tương ứng trong `controller.py` phản hồi lại mail của người gửi.
- `app.py`: tạo giao diện của chương trình gồm hướng dẫn sử dụng và tạo key cho người dùng.
 - Sử dụng thư viện `tkinter` của python để tạo cửa sổ giao diện. Gồm một nút bấm để xuất hiện hướng dẫn sử dụng và một nút bấm để người dùng nhập key và lưu vào trong file `.env` để các file python khác sử dụng.
- `utils.py`: các hàm hỗ trợ thêm cho chương trình.
 - Hàm `get_random_string(length=10)`: trả về một chuỗi ngẫu nhiên.
 - Hàm `recorded()`: dùng để quay video như đã trình bày.
 - Hàm `clean_file_name(text)`: làm sạch tên file, loại bỏ các dấu không cần thiết, chỉ để lại chữ và số.
 - Hàm `time_in_format(format)`: Trả về thời gian theo format mình chọn.
 - Hàm `extract_mail_address(mail_address)`: lấy ra địa chỉ mail.
 - Hàm `mkdir(path)`: tạo một thư mục mới.
 - Hàm `set_registry(REG_PATH, name, value)`: cập nhật giá trị của registry có tên là `name` với giá trị là `value` trong một registry path

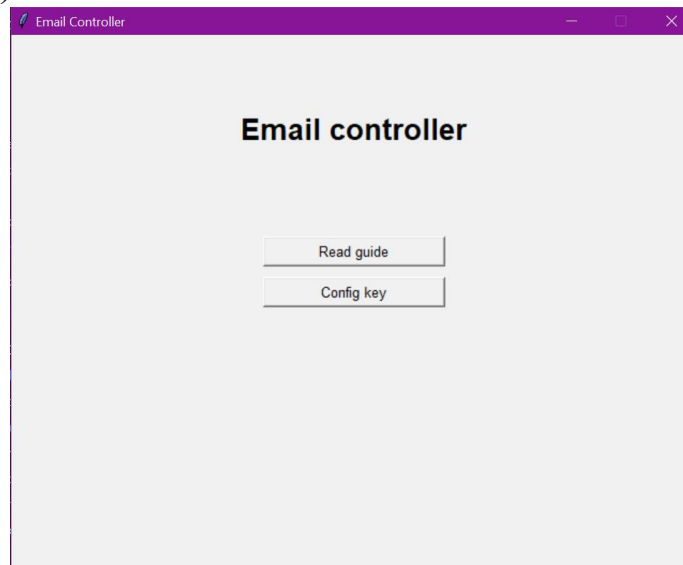
- Hàm `get_registry(REG_PATH, name)`: lấy giá trị của registry có tên là *name* trong một registry path

IV. Cách sử dụng chương trình

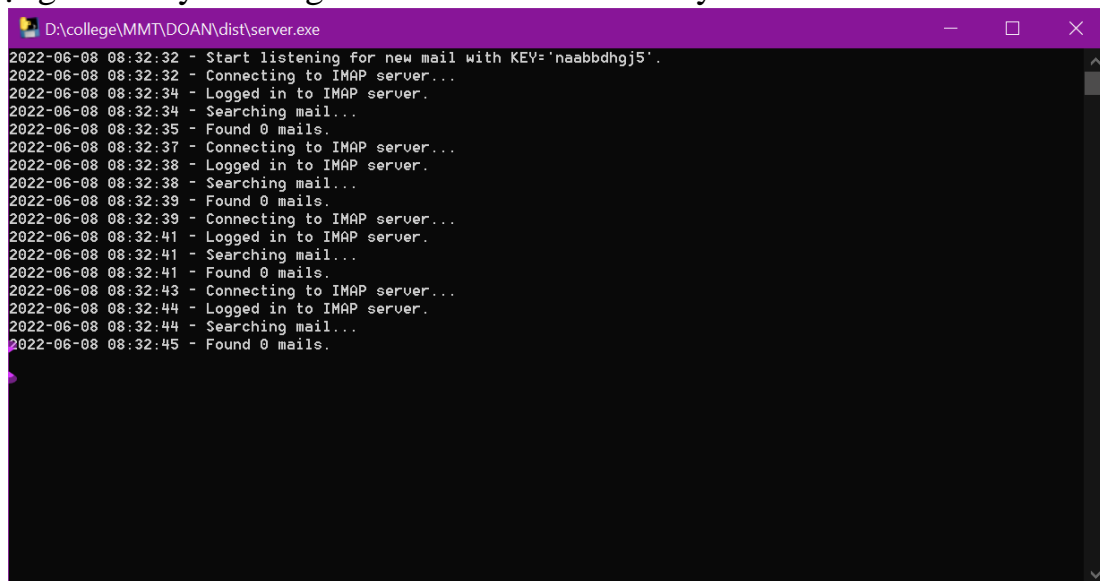
Lưu ý: ứng dụng chỉ sử dụng cho hệ điều hành windows.

1. Tải, cấu hình và chạy chương trình

- Tải thư mục về máy.
- Chạy file `app.exe` để đọc hướng dẫn sử dụng và tạo key cho người dùng (sau đó có thể tắt đi).



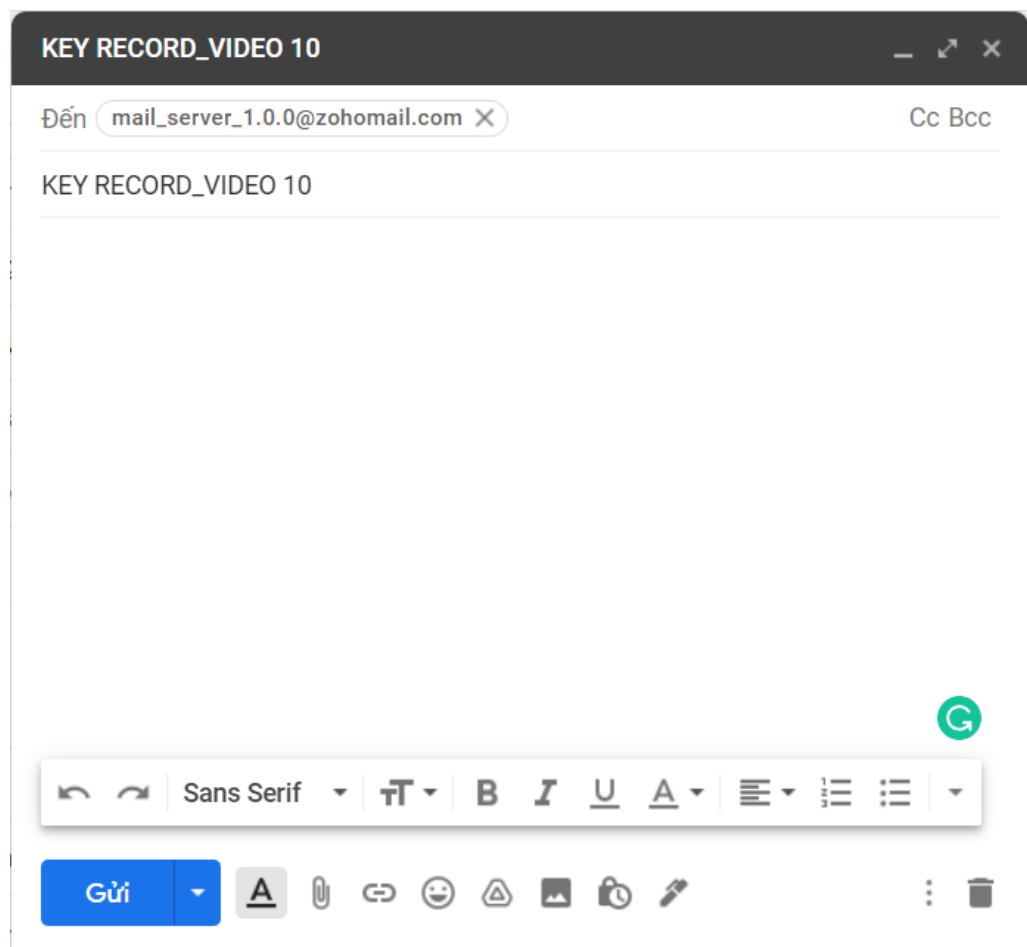
- Chạy file `server.exe`. Lúc này chương trình sẽ bắt đầu chạy và có thể sử dụng. Lúc này có thể gửi mail để điều khiển máy tính.



- Sau khi dùng xong thì tắt file server.exe để kết thúc.

2. Hướng dẫn cú pháp gửi email

- Dùng email cá nhân bất kỳ (Gmail, Yahoo, Outlook, ...).
- Soạn thảo email:
 - Người nhận: mail_server_1.0.0@zohomail.com
 - Chủ đề (subject): <key> <command> <arguments> (nếu có)
 Ví dụ: **KEY RECORD_VIDEO 10**
 Trong đó: “KEY” là key của người dùng, “RECORD_VIDEO” là tên câu lệnh và “10” là tham số, nghĩa là quay trong vòng 10 giây.
 - Nội dung: không cần thiết.



3. Danh sách các câu lệnh:

- SHUTDOWN
- RESTART
- LIST_PROCESS

- STOP_PROCESS_BY_NAME <process_name>
- STOP_PROCESS_BY_ID <process_id>
- SCREENSHOT
- RECORD_VIDEO <duration>
- COPY_FILE <source_path> <dest_path>
- RUN_COMMAND <command>
- CATCH_KEYPRESS <duration>
- GET_REGISTRY <reg_path> <name>
- SET_REGISTRY <reg_path> <name> <value>
- HELP

V. Giải thích một số thuật ngữ

- IMAP: Internet Message Access Protocol là một giao thức để đọc email trên các mail server.
- SMTP: Simple Mail Transfer Protocol là một giao thức để gửi email và định tuyến các mail giữa các mail server.
- MIME: Multipurpose Internet Mail Extensions là một chuẩn được đề xuất bởi Bell Communications năm 1991 để mở rộng những khả năng bị giới hạn của email. MIME là bổ sung cho giao thức SMTP để người dùng có thể gửi non-ASCII như âm thanh, video, hình ảnh,...

VI. Nguồn tài liệu tham khảo

- <https://www.geeksforgeeks.org/e-mail-format/?ref=lbp>
- <https://www.geeksforgeeks.org/multipurpose-internet-mail-extension-mime-protocol/>
- <https://stackoverflow.com/>
- <https://docs.python.org/3/library/imaplib.html>
- <https://docs.python.org/3/library/smtplib.html>
- <https://docs.python.org/3/library/email.html>