

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



KHOA CNTT

* * * * *



Đồ án mạng máy tính

Thành viên nhóm:

Trần Anh Tú

Trần Quang Tuyên

Nguyễn Quốc Vinh

Lớp: 22CTT6

– Tháng 1/2024 –

Mục lục

| | |
|--|----|
| KHOA CNTT | 1 |
| I. Thông tin nhóm | 3 |
| II. Mức độ hoàn thành | 3 |
| 1. Những mục đã làm: | 3 |
| 2. Những mục chưa làm được | 4 |
| III. Kịch bản giao tiếp của chương trình | 4 |
| • SMTP: | 4 |
| • POP3: | 6 |
| 1. Giao thức trao đổi giữa client và server | 8 |
| 2. Cấu trúc thông điệp | 8 |
| 3. Kiểu dữ liệu thông điệp | 8 |
| IV. Hướng dẫn sử dụng các tính năng chương trình | 8 |
| V. Môi trường lập trình và các framework hỗ trợ | 9 |
| 1. Môi trường lập trình | 9 |
| 2. Các Framework được sử dụng | 9 |
| VI. Bảng phân công công việc | 9 |
| 1. Trần Anh Tú | 10 |
| 2. Trần Quang Tuyên | 10 |
| 3. Nguyễn Quốc Vinh | 10 |
| VII. Tài liệu tham khảo | 11 |

I. Thông tin nhóm

- Trần Anh Tú – 22120401
- Trần Quang Tuyên – 22120415
- Nguyễn Quốc Vinh – 22120435

II. Mức độ hoàn thành

❖ Mức độ hoàn thành: 100%

1. Những mục đã làm:

- ✓ Gửi được email To, CC, BCC (xử lý được khi gửi BCC thì những người nhận liên quan (To, CC) sẽ không biết được là có gửi BCC).
- ✓ Gửi được file đính kèm email.

- ✓ Tải được email từ Mail server, không tải những email đã tải trước đó. Xử lý được ngay cả khi tắt/ mở lại Mail Client.
- ✓ Phân loại được email vào các folder cụ thể theo định dạng file config(theo địa chỉ người gửi, theo subject, theo từ khóa trong nội dung email).
- ✓ Mail được tải về client luôn thuộc folder Inbox nếu không phù hợp với kết quả filter nào.
- ✓ Lọc được các từ khóa spam từ subject và nội dung email, xử lý được các email spam vào folder Spam khi lấy email từ Mailbox của Server.
- ✓ Quản lý được trạng thái email đã đọc hay là chưa.
- ✓ Tự động tải email trong mailbox về theo cấu trúc file config.
- ✓ Tổ chức file config có cấu trúc dạng *.json

2. Những mục chưa làm được

- Không có

III. Kịch bản giao tiếp của chương trình

- SMTP:

- Đầu tiên để có thể gửi được Email thì trước hết chúng ta phải tạo một client

- Sau đó chúng ta sẽ gửi lệnh connect đến test mail server với địa chỉ IP của test mail server và port của nó, nếu server trả về 'OK' thì khi đó chúng ta có thể gửi được email đến người nhận.
- Để đúng với format theo yêu cầu respon của server thì chúng ta sẽ xử lý respon của server sau khi nhận lời chào từ client theo đúng format, trong đó chúng ta sẽ xử lý các trường hợp khi client gửi request gì thì server sẽ phản hồi với lệnh tương ứng, ví dụ: client gửi EHLO[IP server] thì server sẽ phản hồi '150 OK' theo đúng format của yêu cầu...
- Khi gửi email thì sẽ có email đến ,To, BCC, tiêu đề email, nội dung email, file đính kèm (có thể có hoặc không), sau đó chúng ta sẽ có nội dung tin nhắn gửi từ client do người dung nhập vào nhập vào, khi gửi thì để tránh người khác có thể đọc được ngoại trừ người đến thì chúng ta sẽ mã hóa(Encode) chúng và gửi đi, sau đó bên nhận sẽ giải mã(Decode) để có thể đọc và hiểu được nội dung gửi.
- Về việc đính kèm file thì chúng ta sẽ viết một hàm set_fileType để chuẩn bị đối tượng MIMEBase để đính kèm và hàm get_filesize để kiểm tra kích thước tệp tin để

đảm bảo rằng tệp tin không quá lớn($\leq 3\text{MB}$) trước khi thực hiện đính kèm

- POP3:

- Server: Gửi lời chào đến client khi thiết lập kết nối

Test Mail Server

- Client: USER <username>

- Trong đó username là tên người dùng của hộp thư ở máy khách ở dạng văn bản thuần túy (không mã hóa)

- Server: gửi mã phản hồi về cho client

- +OK: username là hộp thư hợp lệ
- -ERR: không thể lắng nghe đến hộp thư username

- Client: PASS <string>

- Mật khẩu dành riêng cho máy chủ hộp thư (là chuỗi).

- Server: máy chủ sử dụng user và password đã nhận được để xác định client có quyền truy cập vào hộp thư thích hợp hay không và phản hồi:

- +OK: hộp thư đã khóa và sẵn sàng
- -ERR: mật khẩu sai hoặc không thể khóa thư

- Client: LIST <msg>

- Tham số là số thứ tự (có thể có hoặc không) của hộp thư
- Liệt kê các thư chưa xóa

- Server: gửi mã phản hồi cho client

- +OK và danh sách tin nhắn (số thứ tự cùng kích thước tin nhắn)
- -ERR không có tin nhắn như vậy
- Client: RETR <msg>
 - Tham số là số thứ tự của tin nhắn (bắt buộc)
 - Lấy nội dung thư có số thứ tự tương ứng
- Server: gửi mã phản hồi
 - +OK: máy chủ gửi tin nhắn tương ứng
 - -ERR: không có tin nhắn như vậy
- Client: DELE <msg>
 - Tham số là số thứ tự của tin nhắn (bắt buộc)
 - Đánh dấu tin nhắn có số thứ tự tương ứng sẽ xóa
- Server: gửi mã phản hồi
 - +OK: tin nhắn đã được xóa
 - -ERR: không có tin nhắn nào như vậy
- Client: QUIT
 - Xóa tất cả tin nhắn được đánh dấu
 - Đóng kết nối
- Server: gửi mã phản hồi
 - +OK: đóng kết nối an toàn
 - -ERR: một số tin nhắn không bị xóa

1. Giao thức trao đổi giữa client và server

- Sử dụng giao thức: SMTP, POP3

2. Cấu trúc thông điệp

- *Thông điệp:*

+ SMTP:

- sử dụng lệnh ‘EHLO’ để bắt đầu trao đổi thông điệp với máy chủ
- lệnh ‘DATA’ để bắt đầu quá trình nhập nội dung email
- lệnh ‘QUIT’ để ngắt kết nối giữa client với server

+ POP3:

- lệnh ‘USER’, ‘PASS’ để xác thực đăng nhập
- ‘STAT’ để lấy thông tin về số lượng và kích thước của các email trong hộp thư
- ‘LIST’ để liệt kê các email hoặc ‘RETR’ để nhận một email cụ thể
- ‘DELE’ để đánh dấu tin nhắn sẽ được xóa
- QUIT để xóa toàn bộ tin nhắn được đánh dấu và kết thúc phiên làm việc

3. Kiểu dữ liệu thông điệp

- Kiểu dữ liệu: kiểu văn bản

IV. Hướng dẫn sử dụng các tính năng chương trình

- Đầu tiên chúng ta mở test mail server lên để chờ kết nối từ client.
- Sau đó chúng ta sẽ chạy chương trình trên phần mềm chúng ta viết code, ở trên đó sẽ có menu cho chúng ta chọn như muốn gửi file hay không, To, BCC, ...
- Sau đó chúng ta nhấn send để gửi
- Khi gửi xong thì trên chương trình sẽ báo cho chúng ta là đã gửi thành công, sau đó sẽ có menu tiếp theo để chúng ta chọn và để hỏi chúng ta xem có muốn kết thúc không hay là muốn gửi tiếp hoặc một sự lựa chọn nào khác
- Lưu ý: khi một mục nếu bạn không muốn nhập cái gì khác thì bạn có thể bỏ qua bằng cách nhấn phím 'Enter' như To, BCC,...

V. Môi trường lập trình và các framework hỗ trợ

1. Môi trường lập trình

- Ngôn ngữ lập trình được sử dụng: Python
- Phần mềm được sử dụng để soạn thảo: Visual studio code

2. Các Framework được sử dụng

- Không có quá nhiều framework được sử dụng trong chương trình, chỉ sử dụng thư viện chuẩn của python như: base64, os, email, json, re, time, socket

VI. Bảng phân công công việc

1. Trần Anh Tú

- Lọc các từ khóa spam từ subject và nội dung email, di chuyển các email vào thư mục Spam khi lấy email về từ Mail Box.
- Quản lý được trạng thái email đã đọc hay chưa.

2. Trần Quang Tuyên

- Tải được email từ Mail server, không tải những email đã tải.
- Phân loại các email vào các folder cụ thể theo cấu hình của file config(theo địa chỉ, theo subject hoặc từ khóa), email được tải về thuộc folder Inbox nếu không có kết quả filter nào phù hợp
- Tự động tải email trong mailbox về theo định dạng file config.

3. Nguyễn Quốc Vinh

- Gửi email (dung giao thức SMTP): To, BCC, CC, xử lý được khi gửi BCC thì những người nhận như To, CC sẽ không biết ai được BCC.
- Gửi được file đính kèm.

VII. Tài liệu tham khảo

- [RFC 1939: Post Office Protocol - Version 3 \(rfc-editor.org\)](https://rfc-editor.org/rfc/rfc1939.html)
- [email.parser: Parsing email messages — Python 3.12.1 documentation](https://docs.python.org/3.12.1/library/email.parser.html)
- [email.header: Internationalized headers — Python 3.12.1 documentation](https://docs.python.org/3.12.1/library/email.header.html)
- [Python RegEx \(w3schools.com\)](https://www.w3schools.com/python/python_regex.asp)
- [Python JSON \(w3schools.com\)](https://www.w3schools.com/python/python_json.asp)
- [os.path — Common pathname manipulations — Python 3.12.1 documentation](https://docs.python.org/3.12.1/library/os.path.html)
- [RFC 5321: Simple Mail Transfer Protocol \(rfc-editor.org\)](https://rfc-editor.org/rfc/rfc5321.html)

