

MỤC LỤC

Danh sách hình vẽ	4
Danh sách bảng biểu	5
Danh sách từ viết tắt	6
MỞ ĐẦU	7
PHẦN I: NGUYÊN LÝ HỆ ĐIỀU HÀNH	8
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	8
1. Giới thiệu Deadlock	8
1.1 Định nghĩa	8
1.2 Điều kiện để xảy ra deadlock.....	8
1.3 Các phương pháp xử lý deadlock	9
2. Thuật toán Banker.....	9
CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG	10
1. An toàn trong hệ thống	10
1.1.1 Trạng thái an toàn.....	10
1.1.2 Thứ tự an toàn	10
2. Ký hiệu trong thuật toán Banker.....	10
CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ	11
1. Thuật toán trạng thái an toàn	11
1.1 Dữ liệu vào.....	11
1.2 Quá trình tính toán.....	11
1.2.1 Thuật toán sử dụng.....	11
1.2.2 Độ phức tạp của thuật toán.....	12
1.3 Dữ liệu ra	12
2. Thuật toán yêu cầu tài nguyên	13
2.1 Dữ liệu vào.....	13
2.2 Quá trình tính toán.....	14
2.2.1 Thuật toán sử dụng.....	14
2.2.2 Độ phức tạp của thuật toán.....	14
2.3 Dữ liệu đầu ra	15
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	16

PHẦN II. LẬP TRÌNH MẠNG	17
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	17
1. Email	17
2. Giao thức SMTP	17
3. Giao thức POP3	17
CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG	18
1. Kiểu tổ chức lưu trữ	18
1.1 Tổ chức lưu trữ email.....	18
1.2 Tổ chức lưu trữ thông tin người sử dụng.....	18
2. Hoạt động của hệ thống	18
2.1 SMTP Server	18
2.1.1 Chức năng chính	18
2.1.2 Các lệnh sử dụng.....	19
2.2 POP Server	19
2.2.1 Chức năng chính	19
2.2.2 Các lệnh sử dụng.....	19
2.3 Sơ đồ tổng hợp	19
CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ	21
1. Dữ liệu vào và ra.....	21
2. Triển khai và kết quả	21
2.1 Chức năng đăng nhập.....	21
2.2 Quản lí các mail trong tài khoản.....	22
2.3 Gửi và nhận thư	22
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	23
KẾT LUẬN CHUNG.....	24
TÀI LIỆU THAM KHẢO.....	25
PHỤ LỤC.....	26

PHIẾU ĐÁNH GIÁ CỦA HỘI ĐỒNG DUYỆT ĐỒ ÁN

[illegible]

Danh sách hình vẽ

Hình 1: Ví dụ về deadlock - các ô tô đi qua cầu.....	8
Hình 2: Mẫu định dạng file dữ liệu vào.....	11
Hình 3: Thuật toán tìm kiếm trạng thái an toàn.....	12
Hình 4: Dữ liệu ra trên màn hình console.....	13
Hình 5: Dữ liệu ra trong file	13
Hình 6: Dữ liệu đầu vào cho thuật toán yêu cầu tài nguyên.....	14
Hình 7: Sơ đồ thuật toán yêu cầu tài nguyên.....	14
Hình 8: Dữ liệu đầu ra trong trường hợp yêu cầu hợp lệ.....	15
Hình 9: Dữ liệu đầu ra trong trường hợp yêu cầu lớn hơn nhu cầu thực tế.....	15
Hình 10: Dữ liệu đầu ra trong trường hợp tài nguyên rồi không đủ cung cấp	15
Hình 11: Tổ chức lưu trữ email	18
Hình 12: Sơ đồ hoạt động của SMTP server	18
Hình 13: Sơ đồ hoạt động của POP server	19
Hình 14: Sơ đồ hoạt động tổng quan của hệ thống.....	20
Hình 15: Giao diện màn hình đăng nhập	21
Hình 16: Giao diện màn hình đăng ký tài khoản	21
Hình 17: Giao diện màn hình hộp thư đến.....	22
Hình 18: Giao diện màn hình các thư đã gửi	22
Hình 19: Giao diện màn hình tạo thư mới	22

Danh sách bảng biểu

Bảng 1: Các thành phần chính của một email	17
Bảng 2: Các lệnh sử dụng trong SMTP server	26
Bảng 3: Các lệnh sử dụng trong POP server	27

Danh sách từ viết tắt

STT	Từ viết tắt	Ý nghĩa
1.	P	Tiến trình (Process)
2.	R	Tài nguyên (Resource)
3.	HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
4.	RFC	Request for Comments

MỞ ĐẦU

Mạng là một trong những lĩnh vực đặc thù của ngành Công nghệ thông tin và vấn đề nghiên cứu các đặc điểm, các mô hình và nguyên lý hoạt động của mạng nói chung hay mạng máy tính nói riêng là một trong những yếu tố cần thiết của một sinh viên công nghệ thông tin.

Ngày nay, hệ thống mạng máy tính ngày càng phát triển với nhiều tính năng mới, nhiều ứng dụng mới ngày càng hiện đại, đáp ứng hầu hết các yêu cầu của người dùng từ cá nhân đến doanh nghiệp và các tổ chức lớn khác. Tuy nhiên, để hiểu rõ hơn về bản chất, chúng ta cần đi tìm hiểu những kiến thức cơ bản nhất của nó, các giao thức mà nó sử dụng và quá trình vận hành của một hệ thống mạng là như thế nào.

Chính vì vậy, trong học phần đồ án Cơ sở ngành mạng lần này, em đã quyết định lựa chọn đề tài tương ứng với các phần để hiểu rõ hơn về mạng máy tính cũng như nguyên lý hoạt động của các hệ điều hành như sau:

- *Phần Nguyên lý hệ điều hành:*
Mô phỏng thuật toán nhà băng của Dijkstra để tránh deadlock.
- *Phần Lập trình mạng:*
Tìm hiểu giao thức SMTP và POP3. Ứng dụng xây dựng chương trình gửi và nhận mail.

Em xin chân thành cảm ơn thầy ***ThS. Nguyễn Văn Nguyễn*** và các bạn đã giúp đỡ em trong quá trình thực hiện và hoàn thành đồ án này.

Đà Nẵng, ngày 04 tháng 06 năm 2016

Sinh viên thực hiện

Ngô Trường Phạm Quang

PHẦN I: NGUYÊN LÝ HỆ ĐIỀU HÀNH

TIÊU ĐỀ: MÔ PHỎNG THUẬT TOÁN NHÀ BĂNG CỦA DIJKSTRA ĐỂ TRÁNH DEADLOCK

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1. Giới thiệu Deadlock

1.1 Định nghĩa

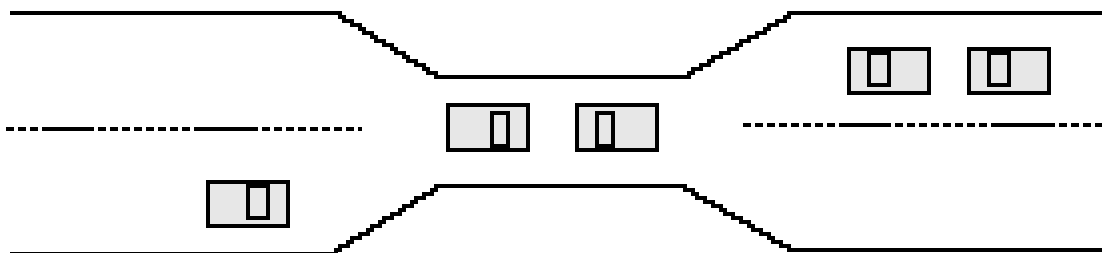
Deadlock là trạng thái xảy ra trong môi trường đa nhiệm (multi-threading) khi hai hoặc nhiều tiến trình có thể cạnh tranh trong việc dùng một số lượng giới hạn tài nguyên. Khi một quá trình yêu cầu tài nguyên nhưng tài nguyên không sẵn có thì quá trình đó sẽ đi vào trạng thái chờ. Quá trình chờ có thể không bao giờ thay đổi lại trạng thái được vì các tài nguyên mà nó yêu cầu bị giữ bởi tiến trình chờ khác.

Khi deadlock xảy ra, các quá trình sẽ không bao giờ hoàn thành việc thực thi và các tài nguyên hệ thống bị buộc chặt, ngăn chặn các quá trình khác bắt đầu.

Ví dụ

Hiện tượng tắc nghẽn trên cầu:

- Hai hay nhiều ô tô đối đầu nhau trên một cây cầu hẹp, chỉ đủ độ rộng cho một chiếc đi qua.
- Mỗi đoạn của cây cầu có thể xem như một tài nguyên.
- Nếu deadlock xuất hiện, nó có thể được giải quyết bằng cách một ô tô lùi lại, nhường đường cho ô tô khác rồi tiến lên sau.



Hình 1: Ví dụ về deadlock - các ô tô đi qua cầu.

1.2 Điều kiện để xảy ra deadlock

Khi bốn điều kiện sau xuất hiện đồng thời

- Điều kiện loại trừ lẫn nhau (mutual exclusion): Một tài nguyên bị chiếm bởi một tiến trình, và không tiến trình nào khác sử dụng tài nguyên này.
- Điều kiện giữ và chờ (hold and wait): Một tiến trình giữ ít nhất một tài nguyên và chờ một số tài nguyên khác rồi để sử dụng.
- Điều kiện không có đặc quyền (preemption): Tài nguyên bị chiếm giữ chỉ có thể rời khi tiến trình “tự nguyện” giải phóng tài nguyên sau khi đã sử dụng xong.

- Điều kiện chờ vòng (circular wait): Một tập tiến trình $\{P_0, P_1, \dots, P_n\}$ có xuất hiện điều kiện “chờ vòng” nếu P_0 chờ một tài nguyên do P_1 chiếm giữ, P_1 chờ một tài nguyên khác do P_2 chiếm giữ, ..., P_{n-1} chờ tài nguyên do P_n chiếm giữ và P_n chờ tài nguyên do P_0 chiếm giữ.

1.3 Các phương pháp xử lý deadlock

Về cơ bản, có ba phương pháp xử lý deadlock như sau:

- Phương pháp 1: Sử dụng một giao thức để hệ thống không bao giờ rơi vào trạng thái deadlock: deadlock prevention (ngăn chặn deadlock) hoặc deadlock avoidance (tránh deadlock).
- Phương pháp 2: Có thể cho phép hệ thống rơi vào trạng thái deadlock, phát hiện deadlock và xử lý nó.
- Phương pháp 3: Bỏ qua deadlock, xem như hệ thống không bao giờ xuất hiện deadlock (có khá nhiều hệ thống sử dụng phương pháp này như Windows, Linux, ...)

Trong phạm vi đề tài, chúng ta sẽ nghiên cứu thuật toán Banker của Dijkstra thuộc phương pháp 1 – tránh deadlock.

2. Thuật toán Banker

Thuật toán đồ thị phân phối tài nguyên có thể sử dụng hiệu quả cho hệ thống với tài nguyên có một thể hiện. Nhưng với những hệ thống mà một tài nguyên có nhiều thể hiện thì thuật toán này không sử dụng được.

Edsger Dijkstra (1930 – 2002), một nhà nghiên cứu khoa học máy tính người Hà Lan đã phát triển thuật toán Banker (trong quá trình thiết kế tiến trình cho hệ điều hành THE) có thể giải quyết được vấn đề nêu trên.

Thuật toán có tên là Banker bởi vì nó có thể được sử dụng trong ngân hàng để đảm bảo rằng ngân hàng không bao giờ rơi vào trạng thái không có tài nguyên. Nếu khách hàng yêu cầu tài nguyên nhưng ngân hàng không thể đáp ứng vì vượt ra ngoài trạng thái an toàn, khách hàng phải chờ cho đến khi có khách hàng khác gửi tiền vào ngân hàng.

CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

1. An toàn trong hệ thống

1.1.1 Trạng thái an toàn

Một trạng thái cấp phát tài nguyên được gọi là an toàn nếu hệ thống có thể cấp phát tài nguyên cho các tiến trình theo một thứ tự nào đó mà vẫn tránh được deadlock.

Hệ thống ở trạng thái an toàn nếu và chỉ nếu tồn tại một thứ tự an toàn.

1.1.2 Thứ tự an toàn

Thứ tự các tiến trình $\langle P_1, P_2, \dots, P_n \rangle$ gọi là một thứ tự an toàn cho trạng thái cấp phát tài nguyên nếu với mỗi P_i , yêu cầu cấp phát của P_i vẫn có thể thỏa mãn căn cứ vào trạng thái của:

- Tất cả các tài nguyên rồi hiện có.
- Tất cả các tài nguyên đang bị chiếm giữ bởi tất cả các P_j với mọi $j < i$.

- **Lưu ý**

- Trạng thái an toàn không là trạng thái bế tắc.
- Trạng thái bế tắc là trạng thái không an toàn.
- Trạng thái không an toàn có thể là trạng thái bế tắc hoặc không.

2. Ký hiệu trong thuật toán Banker

- Tài nguyên rồi: Véc-tơ Available m thành phần. $Available[j] = k$ nghĩa là có k thể hiện rồi của tài nguyên R_j .
- Ma trận yêu cầu tối đa: Ma trận Max $n \times m$ xác định yêu cầu tài nguyên tối đa của mỗi tiến trình. $Max[i][j] = k$ có nghĩa là tiến trình P_i yêu cầu nhiều nhất k thể hiện của tài nguyên R_j để thực hiện.
- Ma trận cấp phát ban đầu: Ma trận Allocation $m \times n$ xác định số thể hiện của các loại tài nguyên đã cấp phát cho mỗi tiến trình. $Allocation[i][j] = k$ có nghĩa là tiến trình P_i được cấp phát k thể hiện của tài nguyên R_j .
- Ma trận nhu cầu: Ma trận Need $n \times m$ chỉ ra số lượng thể hiện của các tài nguyên mà mỗi tiến trình cần cấp phát tiếp để thực hiện. $Need[i][j] = k$ có nghĩa là tiến trình P_i còn cần thêm k thể hiện của tài nguyên R_j nữa.
- Véc-tơ yêu cầu: Request[i] là véc-tơ yêu cầu tài nguyên của tiến trình P_i . $Request[i][j] = k$ có nghĩa là tiến trình P_i yêu cầu k thể hiện của tài nguyên R_j .
- Running là véc-tơ trạng thái thực hiện n thành phần của các tiến trình.
- Quy ước: Nếu hai véc-tơ X, Y thỏa mãn $X[i] \leq Y[i]$ với mọi i thì ta ký hiệu $X \leq Y$.

CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

Sau khi hoàn tất quá trình phân tích và thiết kế hệ thống, chúng ta tiến hành triển khai cài đặt thuật toán và chạy thử nghiệm.

1. Thuật toán trạng thái an toàn

1.1 Dữ liệu vào

Dữ liệu vào bao gồm số lượng tài nguyên, số lượng tiến trình, ma trận cấp phát ban đầu (Allocation), ma trận yêu cầu tối đa (Max) và véc-tơ tài nguyên rỗi (Available) của hệ thống.

Dữ liệu vào có thể được nhập từ bàn phím hoặc đọc từ file. File dữ liệu vào có mẫu như sau:

```
Number Of Processes: 5
Number Of Resources: 4
Allocation Matrix:
0 0 1 2
1 0 0 0
1 3 5 4
0 6 3 2
0 0 1 4
Max Matrix:
0 0 1 2
1 7 5 0
2 3 5 6
0 6 5 2
0 6 5 6
Available Vector:
1 5 2 0
```

Hình 2: Mẫu định dạng file dữ liệu vào

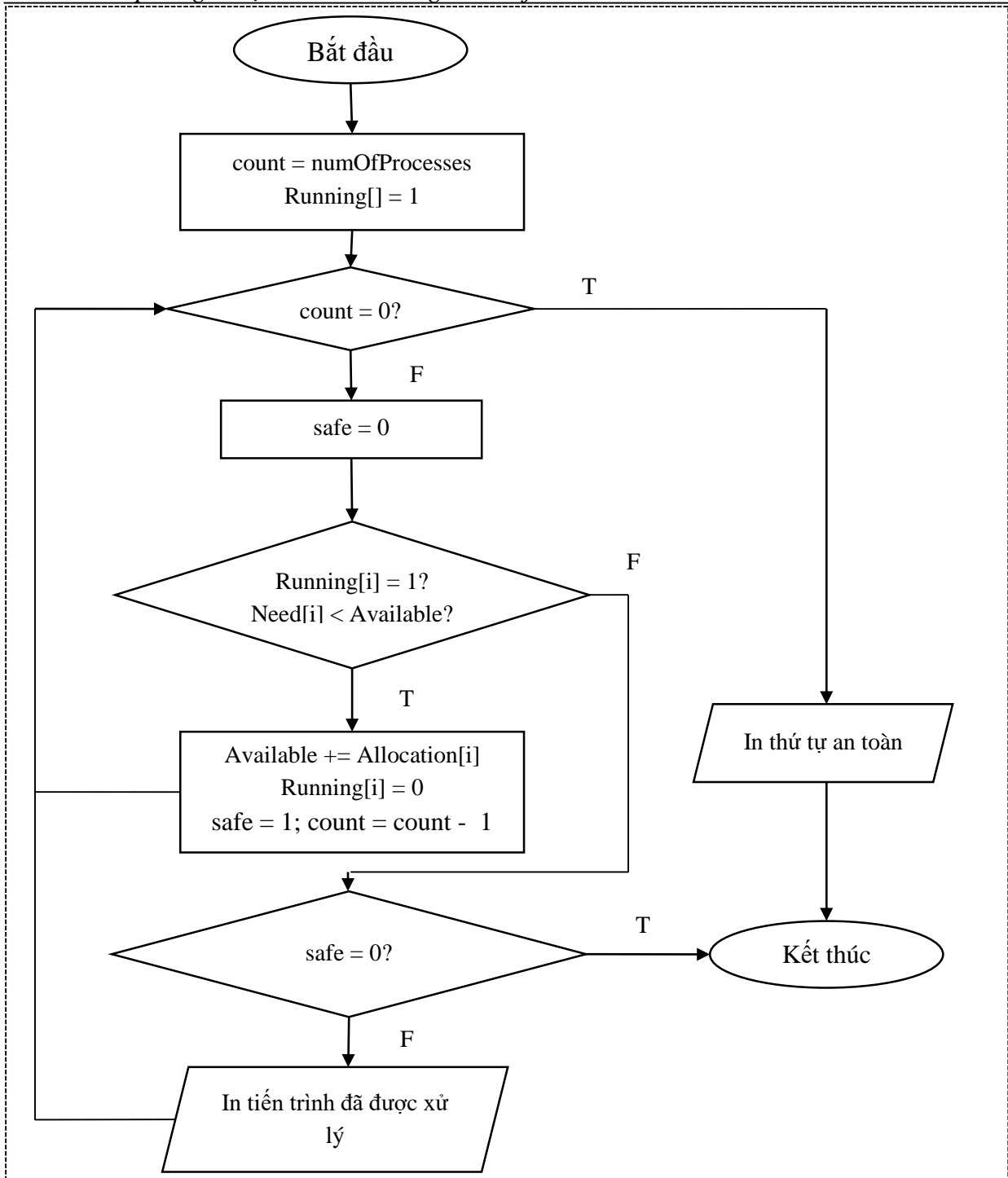
Dữ liệu được đưa vào thử nghiệm chương trình bao gồm:

- Dữ liệu đầu vào hợp lệ, có thể tìm được trạng thái an toàn của hệ thống.
- Dữ liệu đầu vào hợp lệ, không thể tìm được trạng thái an toàn của hệ thống.
- Dữ liệu đầu vào không hợp lệ.

1.2 Quá trình tính toán

1.2.1 Thuật toán sử dụng

Thuật toán sử dụng được mô tả như hình sau.



Hình 3: Thuật toán tìm kiếm trạng thái an toàn

1.2.2 Độ phức tạp của thuật toán

Sau quá trình tính toán và đánh giá, độ phức tạp của thuật toán được tính theo công thức $O(m * n^2)$

Trong đó: m là số lượng tài nguyên của hệ thống.

n là số lượng tiến trình thực hiện trong hệ thống.

1.3 Dữ liệu ra

Chương trình thiết kế để có thể hiển thị dữ liệu đầu ra trên màn hình console cũng như xuất dữ liệu ra file dạng văn bản.

Dữ liệu đầu ra tương ứng với dữ liệu đầu vào như sau:

	Allocation	Max	
P0	0 0 1 2	0 0 1 2	
P1	1 0 0 0	1 7 5 0	
P2	1 3 5 4	2 3 5 6	
P3	0 6 3 2	0 6 5 2	
P4	0 0 1 4	0 6 5 6	(This process is executing.)
Available:	3 14 12 12		
The safe order to execute is: 0 2 1 3 4			
Press any key to continue . . .			

Hình 4: Dữ liệu ra trên màn hình console

```
OUTPUT
Process 0 is executing.
The process is in safe state.
Available vector: 1 5 3 2
Process 2 is executing.
The process is in safe state.
Available vector: 2 8 8 6
Process 1 is executing.
The process is in safe state.
Available vector: 3 8 8 6
Process 3 is executing.
The process is in safe state.
Available vector: 3 14 11 8
Process 4 is executing.
The process is in safe state.
Available vector: 3 14 12 12

The safe order to execute is: 0 2 1 3 4
```

Hình 5: Dữ liệu ra trong file

2. Thuật toán yêu cầu tài nguyên

2.1 Dữ liệu vào

Khi trong hệ thống có một tiến trình muốn yêu cầu tài nguyên, nó phải trải qua các bước kiểm tra của hệ thống bao gồm tài nguyên rồi, tài nguyên sẵn có để cấp phát cho tiến trình đó.

Dữ liệu đầu ban đầu bao gồm các tiến trình trong hệ thống hiện tại và yêu cầu cấp phát tài nguyên của một tiến trình bất kỳ.

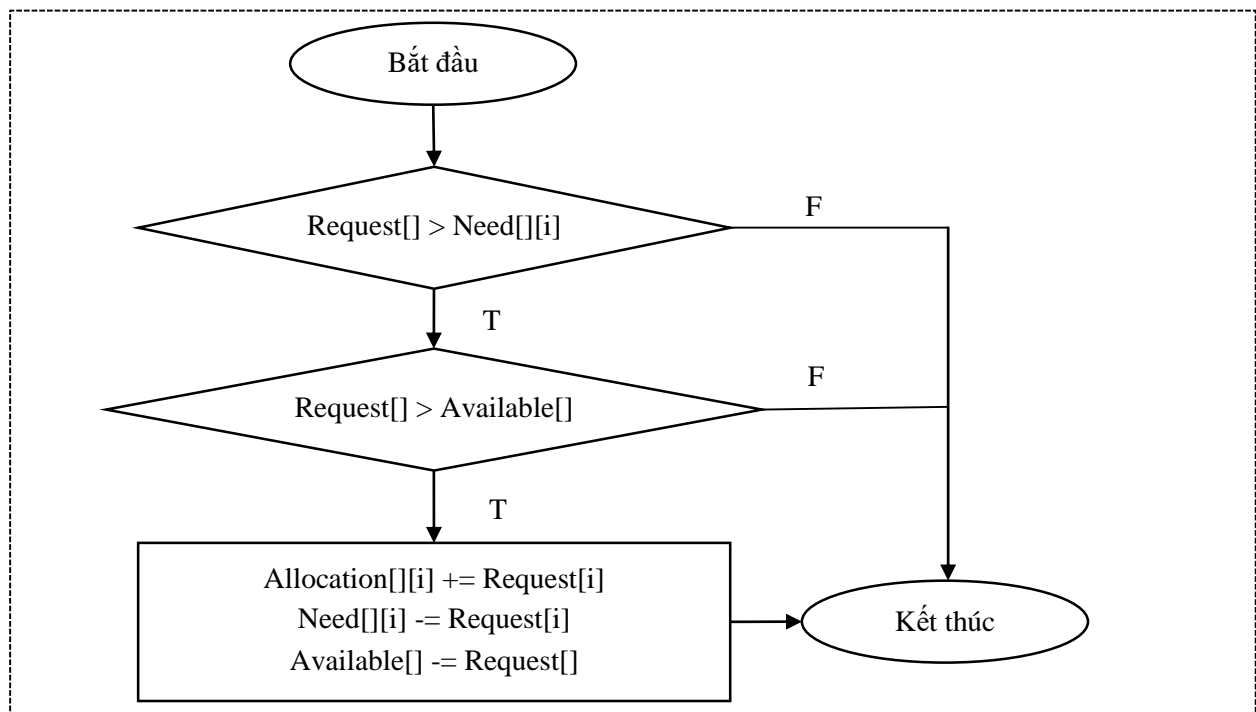
```
Enter your number of selection: 1
The Allocation matrix is:
0 0 1 2
1 0 0 0
1 3 5 4
0 6 3 2
0 0 1 4
The Max Claim matrix is:
0 0 1 2
1 7 5 0
2 3 5 6
0 6 5 2
0 6 5 6
The Need matrix is:
0 0 0 0
0 7 5 0
1 0 0 2
0 0 2 0
0 6 4 2
The Available vector is: 1 5 2 0
The Max Resource vector is: 3 14 12 12
```

Hình 6: Dữ liệu đầu vào cho thuật toán yêu cầu tài nguyên

2.2 Quá trình tính toán

2.2.1 Thuật toán sử dụng

Thuật toán được mô tả như hình sau.



Hình 7: Sơ đồ thuật toán yêu cầu tài nguyên

2.2.2 Độ phức tạp của thuật toán

Độ phức tạp của thuật toán được tính theo công thức $O(m)$ với m là số lượng tài nguyên của hệ thống.

2.3 Dữ liệu đầu ra

Ứng với các dữ liệu đầu vào khác nhau sẽ có các dữ liệu đầu ra khác nhau. Ở đây, chúng ta xét đến 3 trường hợp:

- Dữ liệu đầu vào hợp lệ

```
Your request is accepted.  
The new allocation of process 1 is  
P 1 1 4 2 0  
Continue to check if system? (Y/N)
```

Hình 8: Dữ liệu đầu ra trong trường hợp yêu cầu hợp lệ

- Dữ liệu đầu vào không hợp lệ - Yêu cầu lớn hơn nhu cầu thực tế của tiến trình

```
Enter your process number which requests some resources: 0  
Enter your request vector: 1 1 1 1  
No free resources!
```

Hình 9: Dữ liệu đầu ra trong trường hợp yêu cầu lớn hơn nhu cầu thực tế

- Dữ liệu đầu vào không hợp lệ - Tài nguyên rồi không đủ cung cấp cho yêu cầu của tiến trình

```
Enter your process number which requests some resources: 4  
Enter your request vector: 0 5 3 0  
No available resources can be used!
```

Hình 10: Dữ liệu đầu ra trong trường hợp tài nguyên rồi không đủ cung cấp

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Thuật toán Banker là một trong những thuật toán giúp giải quyết vấn đề deadlock trong một số hệ điều hành. Quá trình nghiên cứu thuật toán đã giúp em hiểu thêm về hiện tượng deadlock trong hệ thống, nguyên nhân gây ra deadlock làm hệ thống bị tê liệt, các phương pháp xử lý tiến trình trong hệ thống cũng như quá trình cấp phát tài nguyên cho các tiến trình.

Chương trình đã đạt được các kết quả như sau:

- Thử nghiệm hết các trường hợp dữ liệu đầu vào theo đề xuất.
- Kết quả đầu ra nhận được thỏa mãn yêu cầu.
- Chương trình chạy nhanh, ổn định.

Các tồn tại của chương trình:

- Giao diện console chưa trực quan và sinh động.
- Chưa thử nghiệm được với các hệ thống lớn với số lượng tài nguyên và tiến trình lớn.

Trong thời gian đến, em sẽ tiếp tục nghiên cứu các thuật toán nhằm ngăn chặn và phòng tránh deadlock khác để so sánh ưu, nhược điểm với thuật toán Banker của Dijkstra. Bên cạnh đó là triển khai thuật toán bằng các ngôn ngữ khác, giúp tạo giao diện đẹp mắt, dễ hiểu và dễ sử dụng hơn cho mọi người.

PHẦN II. LẬP TRÌNH MẠNG

TIÊU ĐỀ: TÌM HIỂU GIAO THỨC SMTP VÀ POP3. ỨNG DỤNG XÂY DỰNG CHƯƠNG TRÌNH GỬI VÀ NHẬN MAIL

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1. Email

Email (electronic mail) là một hệ thống chuyển nhận thư từ qua các mạng máy tính. Một email có thể bao gồm chữ, các nội dung đa phương tiện như hình ảnh, âm thanh, phim hay định dạng HTML.

Email có thể được gửi đi ở dạng mã hóa hay dạng thông thường và được chuyển qua các mạng máy tính, đặc biệt là mạng Internet.

Các thành phần chính của một email bao gồm:

STT	Tên	Ý nghĩa
1.	To:	Địa chỉ mail người nhận
2.	From:	Địa chỉ mail người gửi
3.	Subject:	Tiêu đề email
4.	Content:	Nội dung của email

Bảng 1: Các thành phần chính của một email

2. Giao thức SMTP

SMTP (Simple Mail Transfer Protocol – giao thức truyền tải thư tín đơn giản) là một chuẩn truyền tải thư điện tử qua mạng Internet. SMTP được định nghĩa trong bản RFC 821 (STD 10) và được chỉnh lý bằng bản RFC 1123 (STD 3), chương 5.

SMTP dùng cổng 25 của giao thức TCP để thực hiện quá trình gửi thư điện tử. Trước khi gửi thư, người sử dụng phải chỉ rõ địa chỉ nhận và các địa chỉ này thường được kiểm tra tồn tại trước khi gửi.

3. Giao thức POP3

Giao thức POP (Post Office Protocol) phiên bản 3 (POP3) là một giao thức thuộc tầng ứng dụng, dùng để lấy thư điện tử từ server mail thông qua kết nối TCP/IP, sử dụng cổng mặc định là 110.

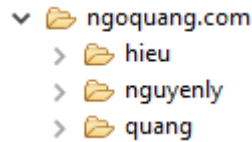
Thiết kế của POP3 hỗ trợ chức năng cho người dùng có kết nối internet không thường trực (như kết nối dial-up), cho phép người dùng kết nối với server, tải mail về, và sau đó có thể xem và thao tác với mail offline. POP3 là một cách thức để nhận các email, trong đó email được lưu giữ trên server để khi người nhận lấy nó. Một khi được nhận bởi người dùng, chúng có thể bị xóa ra khỏi hộp thư của server nếu người dùng lựa chọn.

CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

1. Kiểu tổ chức lưu trữ

1.1 Tổ chức lưu trữ email

Dữ liệu mail gửi/nhận được tổ chức lưu trữ dưới dạng file có phần mở rộng .qms. Mỗi người dùng sẽ có một thư mục lưu trữ trên server chứa toàn bộ mail của người dùng đó.



Hình 11: Tổ chức lưu trữ email

1.2 Tổ chức lưu trữ thông tin người sử dụng

Thông tin người sử dụng được lưu trữ trong file User.txt bao gồm tên đăng nhập và mật khẩu.

2. Hoạt động của hệ thống

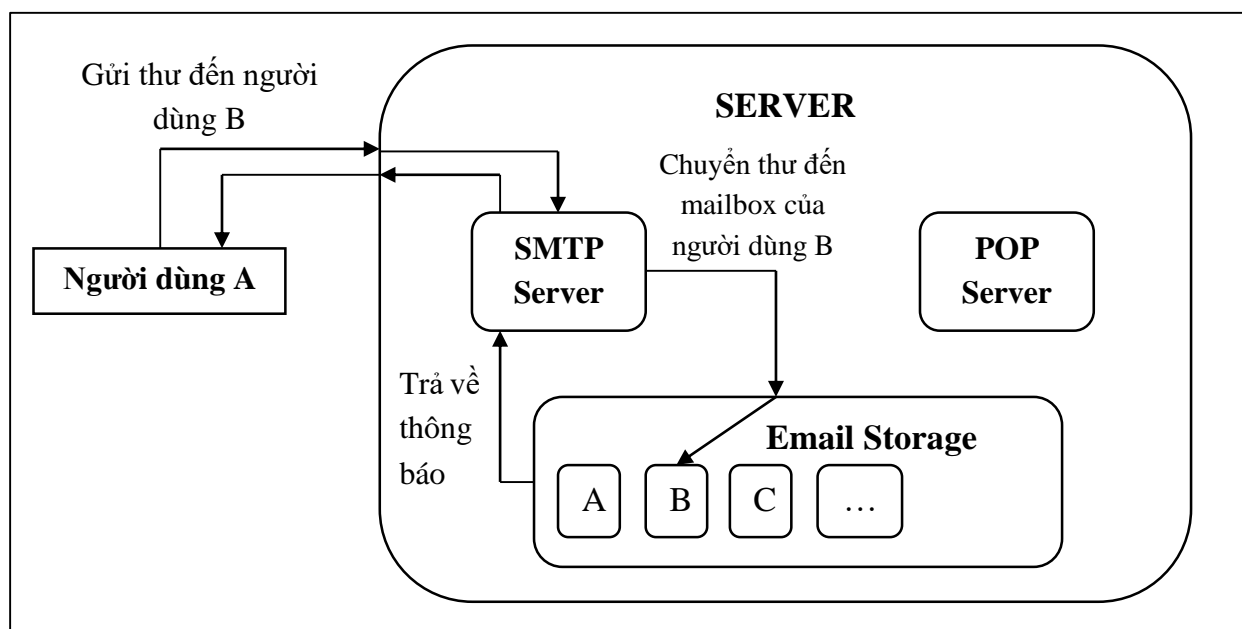
Hệ thống mail server được triển khai thành 2 server riêng biệt là SMTP server và POP server.

2.1 SMTP Server

2.1.1 Chức năng chính

SMTP Server có chức năng chính cơ bản là thực hiện quá trình gửi mail từ tài khoản này qua tài khoản khác. Mail được chuyển trực tiếp tới hộp thư của người nhận và được lưu dưới dạng file.

Quá trình gửi mail bao gồm kiểm tra địa chỉ người nhận, thực hiện gửi mail và trả về kết quả thành công/thất bại.



Hình 12: Sơ đồ hoạt động của SMTP server

2.1.2 Các lệnh sử dụng

SMTP sử dụng theo một bộ tập lệnh chuẩn được định nghĩa trong tài liệu các tài liệu RFC liên quan.

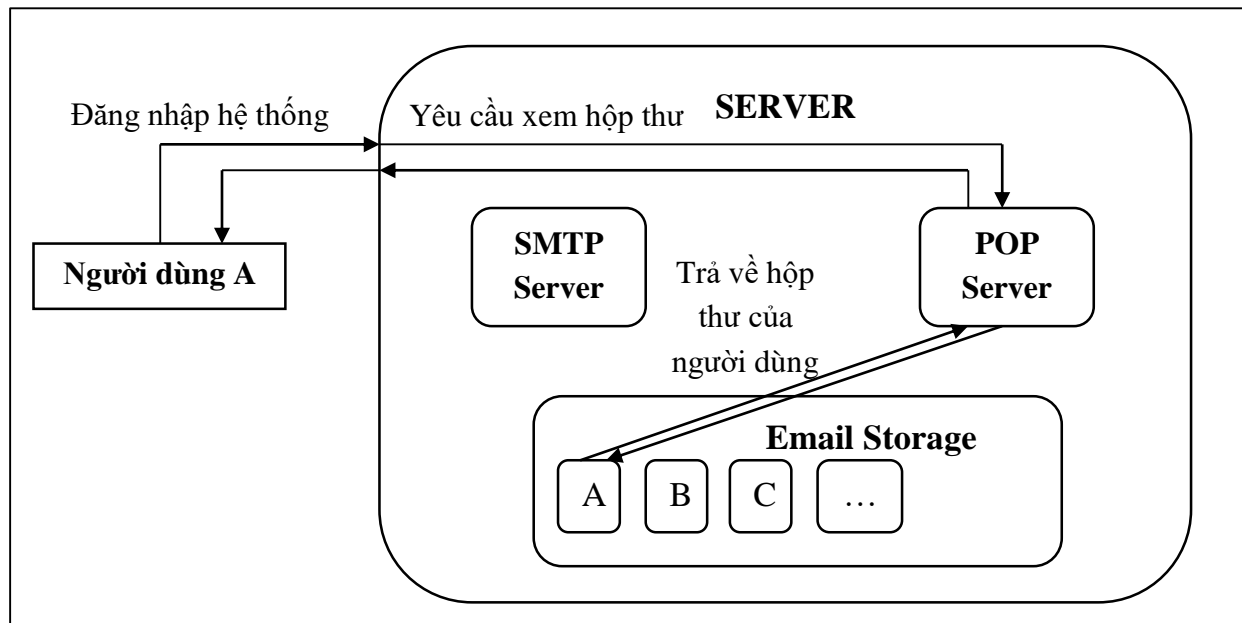
Các lệnh chính phía client bao gồm HELO/ELHO, MAIL FROM, RCPT TO, DATA và QUIT. Server nhận lệnh, xử lý và trả về các mã tương ứng. Chi tiết vui lòng tra cứu tại phần phụ lục.

2.2 POP Server

2.2.1 Chức năng chính

POP Server thực hiện việc kiểm tra người dùng sau khi đăng nhập. Nếu tồn tại tài khoản người dùng trong hệ thống thì nó sẽ trả về hộp thư của người dùng. Nếu người dùng không tồn tại thì POP Server từ chối quyền đăng nhập hệ thống.

Bên cạnh đó, POP Server còn thực hiện các chức năng phụ như xóa thư trong hộp thư khi người dùng lựa chọn xóa từ giao diện của màn hình.



Hình 13: Sơ đồ hoạt động của POP server

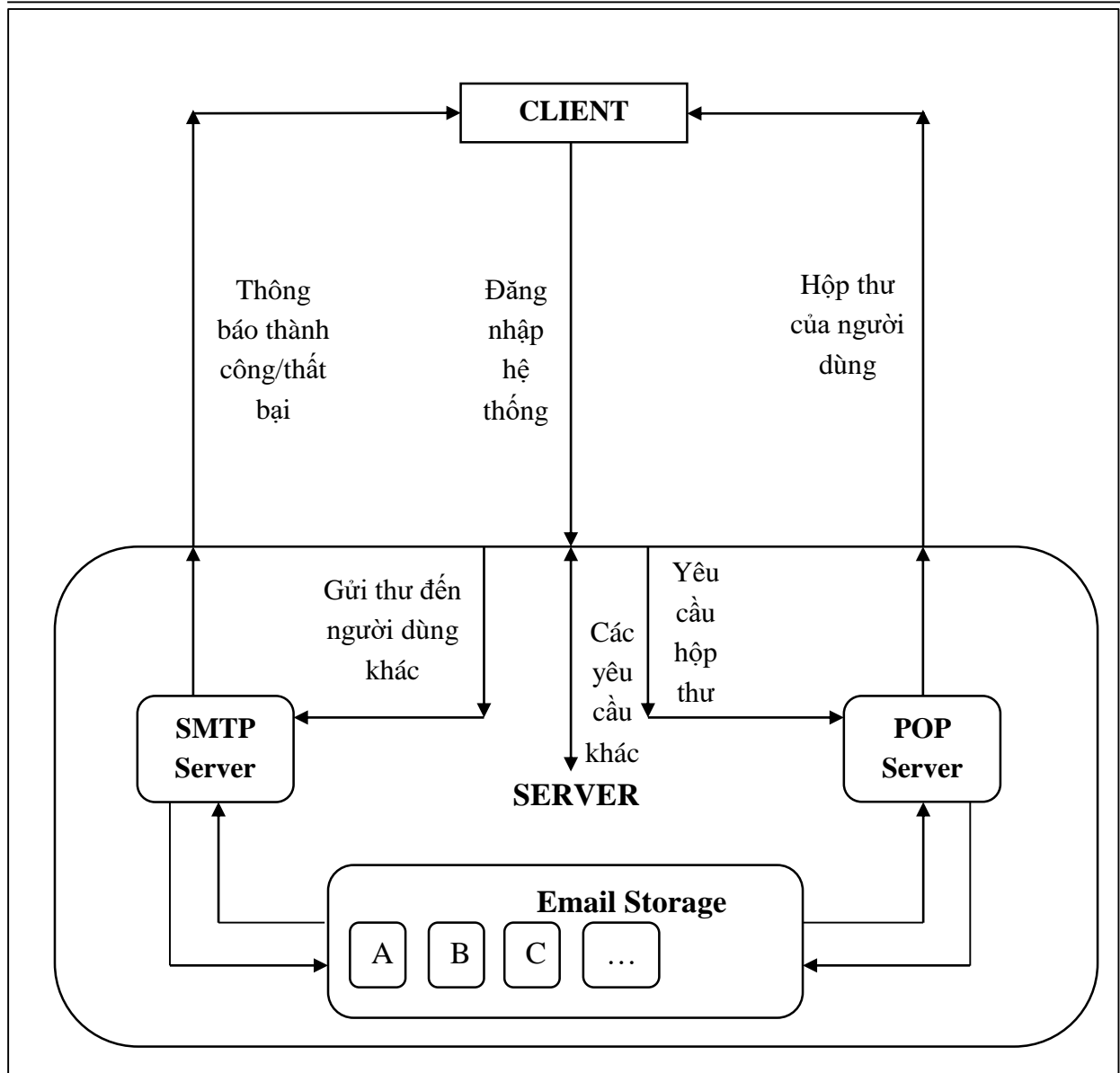
2.2.2 Các lệnh sử dụng

Các lệnh sử dụng chính từ phía client bao gồm USER, PASS, LIST và RETR. Phía server sẽ trả về +OK tương ứng với thành công, và +ERR tương ứng với thất bại kèm thông tin cụ thể. Chi tiết về các lệnh vui lòng tra cứu tại phần phụ lục.

2.3 Sơ đồ tổng hợp

Như vậy, hệ thống gửi, nhận mail hoạt động dựa trên 2 server chính có chức năng riêng biệt, phối hợp với nhau để tạo nên một server mail phục vụ các yêu cầu cơ bản của người dùng.

Hệ thống mail server hoạt động cơ bản theo sơ đồ nguyên lý như hình sau.



Hình 14: Sơ đồ hoạt động tổng quan của hệ thống

CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

1. Dữ liệu vào và ra

Dữ liệu đầu vào của chương trình được nhập từ bàn phím gồm:

- File *User.txt* chứa các thông tin về username và password.
- File *email.properties* chứa các thông tin thuộc tính của mail server
- File có định dạng *.qms* chứa nội dung của các email của người dùng.
- Các file có định dạng khác làm file đính kèm.

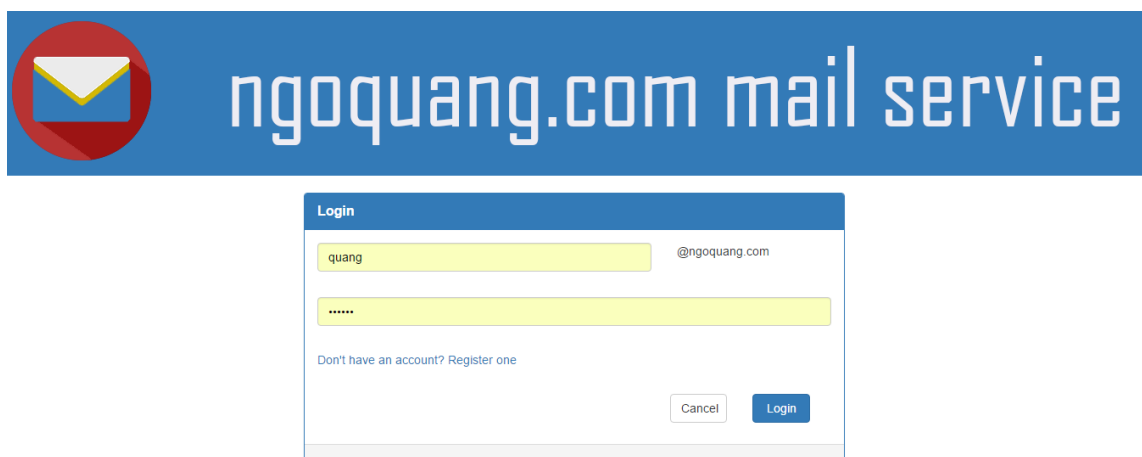
Dữ liệu đầu ra của chương trình là thông tin hiển thị trên màn hình của mail client, gồm nội dung email, tệp đính kèm nếu có.

2. Triển khai và kết quả

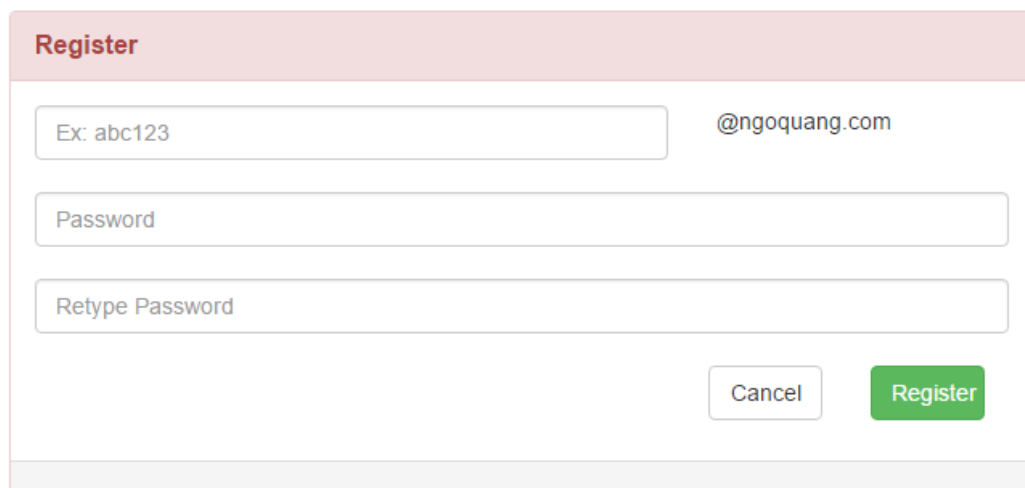
2.1 Chức năng đăng nhập

Người dùng đã có tài khoản của hệ thống mail có thể đăng nhập và sử dụng dịch vụ.

Nếu người dùng chưa có tài khoản có thể chọn chức năng đăng ký để tạo tài khoản mới và bắt đầu sử dụng.



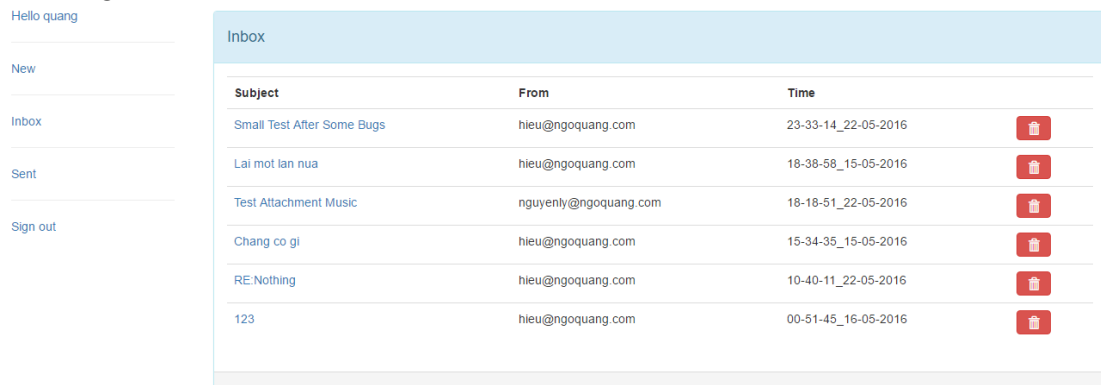
Hình 15: Giao diện màn hình đăng nhập



Hình 16: Giao diện màn hình đăng ký tài khoản

2.2 Quản lí các mail trong tài khoản

Hệ thống được triển khai cho phép người sử dụng có thể quản lí các thư trong hộp thư đến và thư đã gửi. Người dùng chọn vào các chức năng tương ứng ở thanh công cụ bên trái để sử dụng.



Hình 17: Giao diện màn hình hộp thư đến



Hình 18: Giao diện màn hình các thư đã gửi

2.3 Gửi và nhận thư

Đây là chức năng cơ bản nhất của chương trình. Người dùng chọn vào chức năng tạo thư mới, sau đó nhập đầy đủ các thông tin và bấm gửi để gửi thư đến người nhận khác. Khi người nhận đăng nhập vào hệ thống, thư sẽ được đẩy về hộp thư đến của người nhận.

New mail

To:

Subject:

Attachment: No file chosen

Content:

Hình 19: Giao diện màn hình tạo thư mới

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Quá trình triển khai thực hiện đề tài đã giải quyết được các vấn đề sau:

- Tìm hiểu giao thức SMTP, POP3 và các thành phần liên quan.
- Xây dựng được chương trình gửi, nhận mail với các chức năng cơ bản; hoàn thành các yêu cầu mà đề tài đưa ra.
- Giao diện đơn giản, dễ sử dụng.

Các tồn tại của đề tài:

- Chưa có tính bảo mật, mã hóa cho nội dung mail và tài khoản người dùng.
- Thiếu một số chức năng của ứng dụng mail client.

Với những kết quả đạt được cũng như những hạn chế còn tồn tại, em sẽ tiếp tục phát triển ứng dụng mail client để bổ sung thêm các chức năng mới, cải thiện hiệu suất cũng như cập nhật giao diện để đem lại trải nghiệm tốt hơn cho người dùng. Về phía server cần bổ sung thêm các tính năng bảo mật cho hệ thống, tài khoản người dùng và nội dung thư của người dùng. Bên cạnh đó cũng là cải thiện hiệu suất của server, giảm thiểu tối đa tình trạng xuất hiện lỗi và giúp server hoạt động ổn định.

KẾT LUẬN CHUNG

Trong thời gian hơn hai tháng thực hiện đồ án Cơ sở ngành mạng, em đã học hỏi được nhiều kinh nghiệm bằng việc triển khai các kiến thức lý thuyết của các học phần đã học vào thực tế, trong đó chủ yếu là môn Nguyên lý hệ điều hành và Lập trình mạng.

Các kết quả đạt được ở mỗi đề tài giúp em hiểu rõ hơn các nội dung lý thuyết đã học ở trên lớp và ứng dụng các kiến thức đó vào học phần đồ án kỳ này để tạo ra một sản phẩm tương đối hoàn chỉnh.

Trong thời gian đến, em sẽ tiếp tục phát triển thêm các tính năng mới của hai đề tài cũng như khắc phục các hạn chế, tồn tại của mỗi hệ thống. Các giải pháp đã được nêu ra ở mỗi phần kết luận của đề tài sẽ là những cơ sở chính để em phát triển đồ án này.

TÀI LIỆU THAM KHẢO

- [1] **Trần Hồ Thủy Tiên** – Giáo trình Nguyên lý hệ điều hành – Trường Đại học Bách khoa, tháng 02/2016
- [2] **Nguyễn Hải Châu** – Giáo trình Nguyên lý hệ điều hành – Trường Đại học Công nghệ, tháng 02/2016
- [3] **Mai Văn Hà** – Giáo trình Lập trình mạng – Trường Đại học Bách khoa, tháng 04/2016
- [4] **Mai Văn Hà** – Giáo trình Lập trình Java – Trường Đại học Bách khoa, tháng 04/2016
- [5] Trang web: <https://wikipedia.com>, tháng 02/2016
- [6] Trang web: <https://voer.edu.vn>, tháng 02/2016
- [7] Trang web: <https://github.com/oblac/jodd>, tháng 04/2016
- [8] Trang web: http://inetjava.sourceforge.net/lectures/part1_sockets/InetJava-1.8-Email-Examples.html, tháng 04/2016
- [9] Các trang web khác

PHỤ LỤC**1. Các lệnh sử dụng trong SMTP server**

STT	Lệnh từ phía client	Ý nghĩa
1.	HELO/ELHO [domain name]	Gửi lời chào đến server kèm theo domain name của client đang kết nối.
2.	MAIL FROM:<email address>	Xác định địa chỉ người gửi
3.	RCPT TO:<email address>	Xác định địa chỉ người nhận
4.	DATA	Bắt đầu quá trình nhận nội dung thư. Kết thúc nội dung bằng dấu “.”
5.	QUIT	Chấm dứt kết nối với server
Trả lời từ phía SMTP server		
6.	220	Trả về địa chỉ server hiện tại. Server sẵn sàng cho việc gửi mail
7.	250	Thành công
8.	354 End data with <CR><LF>.<CR><LF>	Trả về khi nhận lệnh DATA, bắt đầu quá trình nhận nội dung mail, hướng dẫn người dùng kết thúc gửi thông điệp bằng dấu “.”
9.	503 [Nội dung lỗi]	Thông báo lỗi khi người dùng nhập lệnh sai trình tự cú pháp
10.	500	Lỗi cú pháp hoặc lệnh không xác định
11.	221	Trả về khi người dùng gõ lệnh QUIT, chấm dứt kết nối.

*Bảng 2: Các lệnh sử dụng trong SMTP server***2. Các lệnh sử dụng trong POP server**

STT	Lệnh từ phía client	Ý nghĩa
1.	USER [username]	Yêu cầu đăng nhập vào hệ thống với username được nhập.
2.	PASS [password]	Nhập mật khẩu cho người dùng cần đăng nhập.
3.	STAT	Hiển thị tình trạng của hộp thư của người dùng trên server, bao gồm số lượng thư và tổng dung lượng của các thư.

4.	LIST [message]	Hiển thị danh sách các thư kèm theo kích thước của mỗi thư
5.	RETR n	Hiển thị thư có số thứ tự n khi truy xuất từ hộp thư của hệ thống
6.	DELE n	Đánh dấu thư có số thứ tự n để xóa sau khi đăng xuất khỏi hệ thống
7.	RSET	Hủy bỏ các thao tác vừa thực hiện (như đánh dấu thư để xóa)
8.	QUIT	Chấm dứt kết nối, đăng xuất khỏi hệ thống
	Trả lời từ phía POP server	
9.	+ OK	Thành công
10.	+ ERR	Thất bại

Bảng 3: Các lệnh sử dụng trong POP server