**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN**

**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO BÀI TẬP LỚN**

**LẬP TRÌNH MẠNG (5)**

**ĐỀ TÀI : CHAT CLIENT - SERVER**

Sinh viên thực hiện: Lê Quang Nghĩa

Mã SV : 21IT362

Lớp : 21SE5

Giảng viên hướng dẫn: **ThS. NGUYỄN THANH CẨM**

***Đà Nẵng, 25 tháng 9 năm 2023***

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN**

**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO BÀI TẬP LỚN**

**LẬP TRÌNH MẠNG (5)**

**ĐỀ TÀI : CHAT CLIENT - SERVER**

Sinh viên thực hiện: Lê Quang Nghĩa

Mã SV : 21IT362

Lớp : 21SE5

Giảng viên hướng dẫn: **ThS. NGUYỄN THANH CẨM**

***Đà Nẵng, 25 tháng 9 năm 2023***

LỜI MỞ ĐẦU

Trong thời đại số hóa ngày nay, công nghệ thông tin và việc tin học hoá đang trở thành một trong những yếu tố quyết định quan trọng đối với chính phủ, tổ chức và doanh nghiệp. Chúng đóng vai trò quan trọng trong việc thúc đẩy các bước tiến lớn và mang lại hiệu quả cao.

Sự phát triển nhanh chóng của kỹ thuật máy tính và mạng điện tử đã đánh dấu sự ra đời liên tiếp của các công nghệ mới. Bây giờ, việc truy cập thông tin đã trở nên dễ dàng hơn bao giờ hết, chỉ cần một chiếc điện thoại thông minh kết nối internet và nhập từ khoá liên quan đến vấn đề bạn quan tâm, bạn sẽ có ngay trước mắt toàn bộ thế giới về chủ đề đó, với đầy đủ thông tin, hình ảnh và thậm chí cả âm thanh nếu cần.

Thông qua internet, chúng ta đã thực hiện được nhiều công việc với tốc độ nhanh hơn và chi phí thấp hơn so với các phương thức truyền thống. Một nền tảng phổ biến nhất hiện nay chính là các ứng dụng di động. Có vô số lĩnh vực ứng dụng trên nền tảng di động như quản lý nhân sự, thương mại điện tử, y tế, giáo dục và đào tạo, giải trí và vui chơi, v.v.

Trong học phần này, với mục đích xây dựng chương trình minh họa quá trình tiếp tục và tạm dừng của hệ thống, nhóm chúng em xây dựng thành công một chương trình minh họa các quá trình trong hệ thống.

LỜI CẢM ƠN

Để thực hiện và hoàn thành tốt báo cáo này, chúng em đã nhận được sự giúp đỡ và hướng dẫn rất tận tình của các thầy cô thuộc Khoa Khoa học máy tính –trường đại học Công Nghệ Thông Tin và Truyền Thông Việt Hàn.

Em xin cảm ơn các thầy cô thuộc bộ môn chuyên ngành đã cung cấp cho chúng em các thông tin, kiến thức vô cùng quý báu và cần thiết trong suốt thời gian qua để chúng em có thể thực hiện và hoàn thành báo cáo của mình. Đặc biệt em xin chân thành cảm ơn thầy ThS.Nguyễn Thanh Cẩm, người đã trực tiếp hướng dẫn chúng em trong thời gian thực hiện báo cáo này.

Cuối cùng, xin chân thành cảm ơn các bạn trong ngành công nghệ thông tin đã ủng hộ, giúp đỡ, chia sẻ kiến thức, kinh nghiệm và tài liệu có được giúp chúng tôi trong quá trình nghiên cứu và thực hiện báo cáo. Do giới hạn về mặt thời gian và kiến thức cũng như kinh nghiệm nên báo cáo không tránh khỏi những sai xót. Em rất mong nhận được sự thông cảm của quý thầy cô và mong đón nhận những góp ý của thầy cô và các bạn.

Em xin chân thành cảm ơn!

**NHẬN XÉT CỦA GIẢNG VIÊN**

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

*Đà Nẵng, ngày 03 tháng 10 năm 2023*

Giảng viên hướng dẫn

THS. NGUYỄN THANH CẨM

Contents

[LỜI MỞ ĐẦU 2](#_Toc155940072)

[LỜI CẢM ƠN 3](#_Toc155940073)

[CHƯƠNG 1. TỔNG QUAN VỀ MÔN HỌC 1](#_Toc155940074)

[1.1 Giới thiệu môn học 1](#_Toc155940075)

[1.2 Mục tiêu của môn học 2](#_Toc155940076)

[1.3 Mô tả tóm tắt về học phần. 2](#_Toc155940077)

[1.4 Các điều quan trọng về học phần. 3](#_Toc155940078)

[1.5 Lợi ích của việc học Lập trình mạng. 4](#_Toc155940079)

[CHƯƠNG 2. GIỚI THIỆU VỀ ĐỀ TÀI VÀ CƠ SỞ LÝ THUYẾT 5](#_Toc155940080)

[2.1 Tên đề tài 5](#_Toc155940081)

[2.2 Tổng quan về đề tài 5](#_Toc155940082)

[2.3 Mục tiêu là lý do nghiêm cứu đề tài 6](#_Toc155940083)

[2.4 Giới thiệu về mô hình Client - Server. 6](#_Toc155940084)

[2.5 Giao thức TCP/IP. 7](#_Toc155940085)

[Cấu trúc và chức năng 7](#_Toc155940086)

[2.5.1 IP (Internet Protocol): 8](#_Toc155940087)

[2.5.2 Quá trình truyền tải dữ liệu 8](#_Toc155940088)

[2.5.3 Đặc điểm chính của giao thức 8](#_Toc155940089)

[2.5.4 Sự quan trọng của giao thức 8](#_Toc155940090)

[2.6 Một số ứng ứng dụng mô hình mạng 9](#_Toc155940091)

[2.6.1 Mô hình Client/Server: 9](#_Toc155940092)

[2.6.2 Mô hình Peer to Peer: 11](#_Toc155940093)

[2.7 Giới thiệu về ngôn ngữ lập trình Python 13](#_Toc155940094)

[2.7.1 Python là gì? 13](#_Toc155940095)

[Ưu điểm của Python 13](#_Toc155940096)

[2.7.2 Các công cụ hỗ trợ cho lập trình mạng trong Python 15](#_Toc155940097)

[2.7.2.1 Socket 15](#_Toc155940098)

[2.7.2.2 Threading 15](#_Toc155940099)

[2.7.2.3 Os 16](#_Toc155940100)

[2.7.2.4 PyQt5 16](#_Toc155940101)

[2.7.3 Kết luận 16](#_Toc155940102)

[CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 18](#_Toc155940103)

[3.1 Phân tích yêu cầu 18](#_Toc155940104)

[3.2 Phân tích thiết kế ứng dụng: 19](#_Toc155940105)

[3.2.1 Phía Server 19](#_Toc155940106)

[3.2.1.1 Biểu đồ usecase tổng quát 19](#_Toc155940107)

[3.2.1.2 Đặc tả chức năng 19](#_Toc155940108)

[3.2.1.3 Biểu đồ trình tự 20](#_Toc155940109)

[3.2.2 Phía Client 22](#_Toc155940110)

[3.2.2.1 Biểu đồ usecase tổng quát 22](#_Toc155940111)

[3.2.2.2 Đặc tả chức năng 23](#_Toc155940112)

[3.2.2.3 Biểu đồ trình tự 24](#_Toc155940113)

[CHƯƠNG 4. KẾT QUẢ CHƯƠNG TRÌNH 27](#_Toc155940114)

[4.1 Cấu trúc thư mục 27](#_Toc155940115)

[4.2 Khởi chạy Server 27](#_Toc155940116)

[4.3 Khởi chạy Client 28](#_Toc155940117)

[KẾT LUẬN 31](#_Toc155940118)

# TỔNG QUAN VỀ MÔN HỌC

## Giới thiệu môn học

Môn học "Lập trình mạng" nhằm cung cấp những kiến thức cơ bản và kiến thức sâu hơn về kiến trúc và mô hình mạng máy tính, các giao thức mạng Internet.

Lập trình mạng trang bị cho người học các kiến thức về các mô hình lập trình mạng, nhất là mô hình client/server, mô hình peer-to-peer, mô hình phân tán.

Sinh viên sau khi học xong có thể lập trình với các giao thức TCP/UDP, sinh viên có thể tạo ra các trình duyệt web, web server, mail server, FTP server, chương trình hội thoại,…

* Các kiến thức cơ bản
  + Kiến trúc mạng
  + Lập trình mạng
  + Các loại mạng
  + Hệ điều hành
  + Tập giao thức
  + Các mô hình mạng
  + Mô hình truyền thông trong kiến trúc mạng
  + Các giao thức
  + Mô hình client server
  + Mô hình ứng dụng P2P
  + Lập trình với giao thức TCP
  + Giao thức TCP/IP
  + Lập trình Socket
  + Lập trình với giao thức UDP
  + Lập trình multicast
  + Giao thức HTTP
  + Cơ chế hoạt động Web Server
  + Lập trình RPC
  + Lập trình RMI

## Mục tiêu của môn học

* Về kiến thức

Có kiến thức nền tảng về các kỹ thuật lập trình mạng cơ bản và nâng cao để phát triển các ứng dụng, các dịch vụ chạy trên mạng và mạng Internet.

Biết sử dụng các mô hình mạng để phát triển các ứng dụng mạng như: Peer-to-Peer, Client-Server, mô hình phân tán,…

Biết sử dụng các giao thức trong lập trình mạng như: TCP, UDP, HTTP, MSTP, POP3,…Người học hiểu rỏ hơn nguyên lý hoạt động của các giao thức trên.

* Về kỹ năng :

Có kỹ năng lập trình mạng, lập trình đa tuyến, lập trình phân tán. Kỹ năng xây dựng các ứng dụng mạng và đóng gói, xây dựng các dịch vụ mạng phổ biến trên Internet.

Có kỹ năng làm việc độc lập, làm việc nhóm, kỹ năng thuyết trình, kỹ năng đọc tài liệu chuyên ngành tiếng nước ngoài ….

* Về thái độ :

Nhận thức được các chuẩn mực và văn hóa trong môi trường lập trình mạng để có thể tham gia phát triển các dự án phần mềm chuyên nghiệp. Định hướng được vị trí việc làm thông qua việc hoàn thành các đồ án của môn học.

## Mô tả tóm tắt về học phần.

Học phần gồm các nội dung chính như sau: Giới thiệu khái quát các kiến trúc và mô hình mạng, mô hình OSI, mô hình TCP/IP, mô hình Client-Server,…. Các giao thức phổ biến trong lập trình mạng như: TCP, UDP, HTTP, FTP, POP3,…. Các kỹ thuật lập trình Multicasst, RMI, SOA. Xây dựng các ứng dụng như: web server, mail server, FTP server

## Các điều quan trọng về học phần.

Mục tiêu của lập trình mạng: Lập trình mạng nhằm phát triển các ứng dụng và dịch vụ có khả năng giao tiếp qua mạng. Điều này bao gồm việc truyền tải dữ liệu, thiết lập kết nối, xử lý giao tiếp giữa các máy tính và các thiết bị trong mạng.

Giao thức mạng: Để truyền thông qua mạng, lập trình mạng thường sử dụng các giao thức mạng như TCP/IP, UDP, HTTP, FTP, và nhiều giao thức khác. Các giao thức này định rõ cách dữ liệu được đóng gói, gửi và nhận trên mạng.

Socket Programming: Một phần quan trọng của lập trình mạng là sử dụng socket, một cơ chế cho phép các máy tính trò chuyện và truyền dữ liệu qua mạng. Lập trình socket là cách cơ bản để xây dựng ứng dụng mạng.

Các ngôn ngữ lập trình mạng: Các ngôn ngữ phổ biến để lập trình mạng bao gồm C, C++, Python, Java, Ruby, và nhiều ngôn ngữ khác. Mỗi ngôn ngữ cung cấp thư viện và API để tạo ứng dụng mạng.

Các loại ứng dụng mạng: Lập trình mạng có thể được áp dụng vào nhiều loại ứng dụng, bao gồm ứng dụng web (web development), ứng dụng máy chủ (server applications), ứng dụng trò chuyện (chat applications), trò chơi trực tuyến (online gaming), IoT (Internet of Things), và nhiều ứng dụng khác.

Bảo mật mạng: An ninh mạng (network security) là một khía cạnh quan trọng của lập trình mạng. Phải có các biện pháp để đảm bảo tính bảo mật và bảo vệ dữ liệu khi truyền qua mạng.

Debugging và Testing: Việc kiểm tra và gỡ lỗi (debugging) trong lập trình mạng thường khá phức tạp do tình hình mạng thay đổi và các vấn đề về đồng thời hóa (concurrency). Cần phải sử dụng các công cụ và phương pháp đặc biệt để kiểm tra và sửa lỗi.

Phát triển ứng dụng đa nền tảng: Lập trình mạng có thể được thực hiện trên nhiều nền tảng khác nhau, bao gồm ứng dụng máy tính, ứng dụng di động, và ứng dụng web.

## Lợi ích của việc học Lập trình mạng.

Giúp sinh viên hiểu rõ kiến trúc và nguyên lý hoạt động của các mô hình mạng thông dụng.

Sử dụng được các mô hình peer-to-peer, client/server, mô hình phân tán trong lập trình mang, sử dụng được các giao thức TCP, UDP, HTTP, FTP, POP3 trong lập trình mạng để xây dựng các ứng dụng chạy trên môi trường internet, phân tích được nguyên lý hoạt động của các tiến trình, các giao thức trong kỹ thuật lập trình mạng.

Xây dựng được các chương trình hội thoại, trình duyệt, web server, mail server, FTP server.

# GIỚI THIỆU VỀ ĐỀ TÀI VÀ CƠ SỞ LÝ THUYẾT

## Tên đề tài

* Tên đè tài : Chat Client – Server.

## Tổng quan về đề tài

Chat client-server là một hình thức giao tiếp giữa hai hoặc nhiều máy tính thông qua mô hình client-server, trong đó một máy tính hoạt động như server, chịu trách nhiệm quản lý và cung cấp dịch vụ, trong khi các máy tính khác kết nối như client để sử dụng các dịch vụ đó.

Trong trường hợp của chat client-server, mục tiêu chính là truyền thông giữa các người dùng qua mạng.

Ứng dụng chat client-server cung cấp một giao diện cho người dùng để gửi và nhận tin nhắn, hình ảnh, và có thể thậm chí là tệp tin.

Server đóng vai trò quản lý, duy trì danh bạ người dùng, và đảm bảo truyền thông an toàn và hiệu quả.

Đối với client, nó chủ yếu là giao diện người dùng, nơi người dùng có thể tương tác với hệ thống để gửi và nhận thông điệp.

Các ứng dụng chat client-server rất phổ biến trong nhiều lĩnh vực, từ trò chuyện cá nhân, doanh nghiệp, đến các ứng dụng xã hội.

Chúng đem lại khả năng giao tiếp trực tuyến nhanh chóng và thuận tiện, đồng thời cung cấp môi trường an toàn và bảo mật thông tin.

Các tính năng như nhóm chat, gọi video, và chia sẻ tệp tin thường được tích hợp để tối ưu hóa trải nghiệm người dùng.

Ngoài ra, chat client-server còn đặt ra những thách thức về bảo mật, quản lý tài nguyên, và khả năng mở rộng để đáp ứng nhu cầu ngày càng lớn.

Việc thiết kế một hệ thống hiệu quả đôi khi đòi hỏi sự cân nhắc kỹ lưỡng về cấu trúc mạng, giao thức truyền thông, và các chiến lược quản lý dữ liệu.

## Mục tiêu là lý do nghiêm cứu đề tài

* Thiết kế kiến trúc của hệ thống
* Xây dựng mô hình hoạt động của Hệ thống chat
* Tìm hiểu, lựa chọn các giao thức mạng trong Hệ thống Chat.
* Thiết kế được hệ thống Chat đơn giản: đăng nhập – đăng xuất khỏi hệ thống chat, trao đổi thông tin (chat) giữa các User trong hệ thống
* Tích hợp các chức tính năng trong Hệ thống Chat: chat riêng, chat nhóm, gửi file, ảnh, lưu trữ dữ liệu người dùng

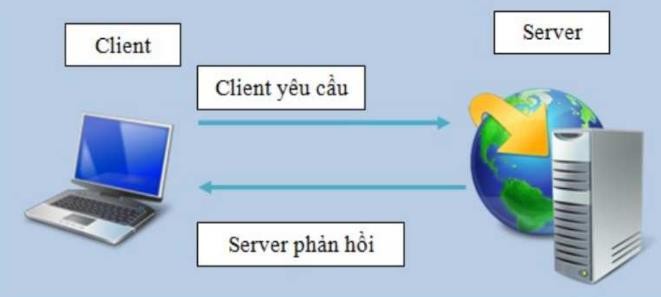
## Giới thiệu về mô hình Client - Server.

Server được hiểu là máy chủ, thường là một hệ thống máy lớn, có bộ xử lý mạnh, có khả năng hoạt động đáng tin cậy, có khả năng luu trữ dữ liệu lơn, nó chuyên quản lý tài nguyên ( chủ yếu là cơ sở dữ liệu), cung cấp các dịch vụ mạng cho các máy khách hàng (client) sử dụng. Bình thường nó chạy suốt thời gian thực và sẳn sàng chấp nhận các yêu cầu kết nối và các yêu cầu dịch vụ khác từ máy khách (Client).

Client là máy khách, nó thường được sử dụng bởi người dùng cuối. Nó hoạt động dựa trên việc sử dụng dihj vụ mà máy server cung cấp để thực hiệ các hoạt dộng dựa trên việc sử dụng dịch vụ mà máy server cung cấp để thực hiện các cồn việc mà người dùng cuối mong muốn.

Quá trình hoạt động của mô hình này là lặp lại 2 quá trình sau:

* Client gửi yêu cầu lên server.
* Server nhận được yêu cầu thì sẽ xử lý và phản hồi lại client.



***Hình : Quy trình hoạt động mô hình Client - Server***

* Mỗi cổng trong mỗi thời điểm được sử dụng cho tối đa là 1 ứng dụng. Ví dụ như một ứng dụng nào đó đã sử dụng cổng 55 thì ứng dụng khác không thể sử dụng cổng 55 đó nữa chùng nào ứng dụng trước đó đóng cổng 55 lại.

## Giao thức TCP/IP.

Giao thức TCP/IP (Transmission Control Protocol/Internet Protocol) là bộ giao thức tiêu chuẩn được sử dụng rộng rãi trong môi trường mạng, đặc biệt là trên Internet. Được thiết kế để cung cấp giao tiếp hiệu quả và tin cậy giữa các thiết bị khác nhau, giao thức này đóng vai trò quan trọng trong việc kết nối và truyền tải dữ liệu.

Cấu trúc và chức năng

TCP (Transmission Control Protocol):

* Là giao thức quản lý truyền tải dữ liệu có độ tin cậy cao.
* Thiết lập và duy trì kết nối giữa các thiết bị.
* Đảm bảo gói tin dữ liệu được chuyển đến đúng đích mà không bị mất mát hoặc bị lỗi.

### IP (Internet Protocol):

* Xác định địa chỉ và định danh cho các thiết bị trong mạng.
* Quản lý việc định tuyến dữ liệu từ nguồn đến đích.
* Hỗ trợ việc truyền tải dữ liệu qua các mạng khác nhau.

### Quá trình truyền tải dữ liệu

3-Way Handshake:

* Giai đoạn bắt đầu kết nối giữa client và server.
* Bao gồm ba bước: SYN, SYN-ACK, ACK. Duy trì Kết Nối:
* Cung cấp cơ chế duy trì kết nối liên tục giữa hai thiết bị trong suốt quá trình truyền tải dữ liệu.

Đóng Kết Nối:

* Kết thúc quá trình truyền tải dữ liệu một cách an toàn và đảm bảo.

### *Đặc điểm chính của giao thức*

Tính Tin Cậy: Đảm bảo tin cậy trong truyền tải dữ liệu thông qua cơ chế xác nhận và điều khiển lỗi.

Đa Nhiệm: Hỗ trợ việc truyền tải đồng thời nhiều luồng dữ liệu giữa các thiết bị.

Độc Lập với Ngữ Cảnh: Hoạt động độc lập với ứng dụng và hệ điều hành, giúp kết nối nhiều hệ thống khác nhau.

### Sự quan trọng của giao thức

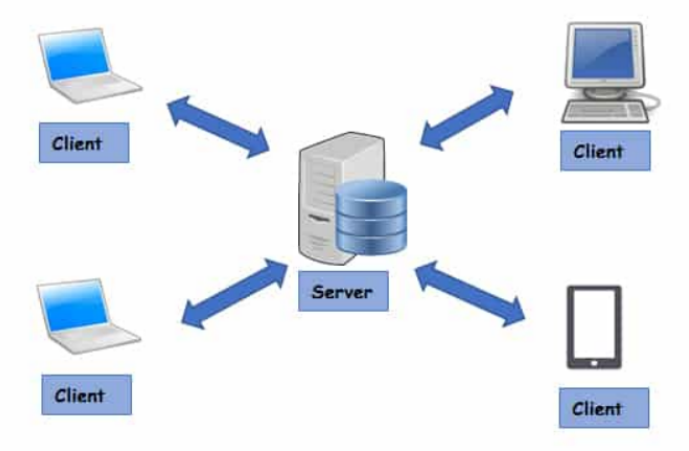
Giao thức TCP/IP có một vai trò quan trọng trong việc đảm bảo giao tiếp hiệu quả và đáng tin cậy giữa các ứng dụng mạng. Sự linh hoạt, tin cậy và tính tương thích của nó đã giúp giao thức này trở thành tiêu chuẩn trong quá trình phát triển các ứng dụng kết nối mạng, bao gồm cả ứng dụng chat theo mô hình client-server.

Thông qua sự hiểu biết về giao thức TCP/IP, chúng ta có cơ sở để xây dựng và triển khai chương trình chat với tính năng ổn định và đáng tin cậy, giúp tối ưu hóa trải nghiệm người dùng trong môi trường mạng.

## Một số ứng ứng dụng mô hình mạng

### Mô hình Client/Server:

Client/Server là mô hình mạng máy tính bao gồm 2 thành phần chính là máy khách (client) và máy chủ (server). Trong mô hình này, server là nơi lưu trữ tài nguyên, cài đặt các chương trình dịch vụ và thực hiện các yêu cầu của client. Client đón vai trò gửi yêu cầu đến server. Client gồm máy tính và thiết bị điện tử nói chung.



Hình : Mô hình Client- Server

**Ưu nhược điểm của Client server:**

+ Ưu điểm:

* Giao thức này không phải chịu bất kỳ một sự kiểm soát nào đến từ các tổ chức nào. Chính vì điều này tạo ra cho người dùng sự tự do và thoải mái trong khi sử dụng.
* Giao thức TCP/IP này là loại giao thức có khả năng tương thích cao trên thị trường đối với tất cả các hệ điều hành, trên cả các phần cứng của thiết bị nên việc giao thức này hiển nhiên được hoạt động hiệu quả nhất trên nhiều những hệ thống khác nhau.
* Cuối cùng giao thức này có được một khả năng mở rộng cao, có thể thực hiện việc định tuyến nên khi thông qua mạng chúng có thể tự mình xác định được những đường dẫn hiệu quả nhất.

**Nhược điểm**

* Giao thức này là một giao thức rất khó để quản lý chúng, cũng như việc cài đặt để sử dụng được giao thức này khá phức tạp.
* Điều khó khăn tiếp theo của giao thức này là chúng không dễ để có thể thay thế được.
* Ngoài ra trên tầng transport của giao thức này cũng không đảm bảo cho việc phân phối được các gói tin.
* TCP/IP cũng không được hiệu quả khi mô tả các công nghệ trong hệ thống mạng mới mới do không thể tách biệt rõ ràng được giữa những khái niệm về dịch vụ, các giao diện và với giao thức.
* Nhược điểm cuối cùng của giao thức này là dễ bị tấn công SYN, đây là một kiểu tấn công từ chối các dịch vụ.

Dù nhiều nhược điểm là thế nhưng giao thức này lại là giao thức được ưa chuộng sử dụng nhất, vì những ưu điểm nổi bật của chúng khiến cho người dùng ít bận tâm đến những nhược điểm này.

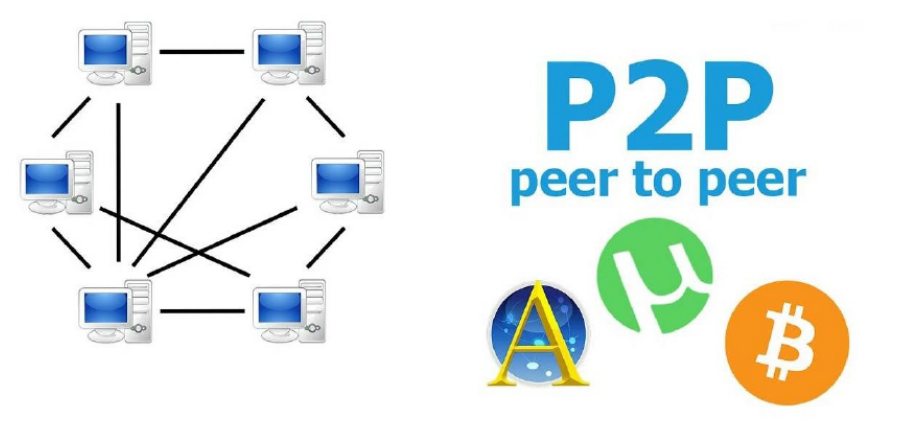
**+ Nguyên lý hoạt động của Client/Server:**

Client hay chính là máy khách, máy trạm – là nơi gửi yêu cầu đến server. Nó tổ chức giao tiếp với người dùng, server và môi trường bên ngoài tại trạm làm việc. Client tiếp nhận yêu cầu của người dùng sau đó thành lập các query string để gửi cho server. Khi nhận được kết quả từ server, client sẽ tổ chức và trình diễn những kết quả đó.

Server xử lý yêu cầu gửi đến từ client. Sau khi xử lý xong, server sẽ gửi trả lại kết quả, client có thể tiếp tục xử lý các kết quả này để phục vụ người dùng. Server giao tiếp với môi trường bên ngoài và client tại server, tiếp nhận yêu cầu dưới dạng query string (xâu ký tự). Khi phân tích xong các xâu ký tự, server sẽ xử lý dữ liệu.

### Mô hình Peer to Peer:

P2P – Peer-to-peer (mạng ngang hàng) là một mô hình mạng phi tập trung với các bên có các cấu trúc phiên giao tiếp giống nhau. Trong đó, mỗi nút hoạt động giống như một máy khách và máy chủ của hệ thống cho phép chia sẻ các phương tiện truyền thông với nhau dễ dàng hơn, nhanh chóng hơn.



*Hình : Mô hình P2P*

**Ưu điểm và nhược điểm của P2P**

* **Ưu điểm:**

Cấu trúc của mạng Peer to Peer được duy trì bởi người dùng của nó, những người có thể vừa cung cấp và sử dụng tài nguyên. Không có cái gọi là máy chủ trung tâm hoặc máy chủ lưu trữ, điều này làm cho các hệ thống P2P rất khác với các mô hình máy khách - máy chủ truyền thống, nơi dữ liệu được phân phối một chiều (từ máy chủ tập trung đến các máy khách của nó), vì cấu trúc này mà mạng P2P đem lại một số ưu điểm sau cho người dùng:

* Mạng Peer to Peer **không cần hệ điều hành** mạng.
* **Không cần một máy chủ đắt tiền** vì các máy trạm riêng lẻ được sử dụng để truy cập các tệp dữ liệu.
* **Không cần nhân viên chuyên môn** như kỹ thuật viên mạng.
* **Dễ thiết lập** hơn nhiều so với mạng máy khách - máy chủ, không cần kiến ​​thức chuyên môn nhiều.
* Nếu một máy tính bị lỗi, nó sẽ **không làm gián đoạn** bất kỳ phần nào khác của mạng. Nó chỉ có nghĩa là những tệp đó không có sẵn cho những người dùng khác tại thời điểm đó.
* **Nhược điểm của mạng P2P**

Mặc dù có những ưu điểm trên nhưng mạng P2P cũng có những nhược điểm riêng:

* Vì mỗi máy tính có thể bị người khác truy cập nên có thể làm **chậm hiệu suất** của người dùng.
* Các tệp và thư mục **không thể được sao lưu tập trung**.
* Các tệp và tài nguyên **không được tổ chức tập trung** vào một “khu vực được chia sẻ” cụ thể. Chúng được lưu trữ trên các máy tính riêng lẻ và có thể khó xác định nếu chủ sở hữu máy tính không có hệ thống lưu trữ hợp lý.
* Nếu vi rút xâm nhập mạng sẽ có rất **nhiều người dùng bị ảnh hưởng**. Đảm bảo rằng vi rút không được đưa vào mạng là trách nhiệm của mỗi người dùng cá nhân.
* Các **tài nguyên sẽ biến mất** vì node cung cấp tài nguyên bị ngắt kết nối bất cứ lúc nào.

## Giới thiệu về ngôn ngữ lập trình Python

### Python là gì?

Python là một ngôn ngữ lập trình thông dịch, dễ đọc và dễ hiểu, được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm lập trình mạng. Được tạo ra với triết lý "đọc được, đơn giản và dễ sử dụng," Python là sự lựa chọn lý tưởng cho các dự án mạng và ứng dụng chat.



***Hình 2: Logo ngôn ngữ lập trình Python***

Ưu điểm của Python

Dễ Đọc và Dễ Hiểu:

+ Cú pháp của Python được thiết kế để giống ngôn ngữ tự nhiên, tạo ra mã nguồn dễ đọc và dễ hiểu. Điều này giúp giảm độ phức tạp của mã nguồn, làm cho quá trình phát triển và bảo trì ứng dụng trở nên thuận tiện hơn. Nhờ vào việc sử dụng từ ngữ đơn giản và cấu trúc mã nguồn rõ ràng, Python giúp nhà phát triển dễ dàng theo dõi luồng logic của chương trình và giảm khả năng phạm lỗi do hiểu lầm cú pháp.

Tính Linh Hoạt và Đa Nhiệm:

+ Python là một ngôn ngữ linh hoạt và đa nhiệm, có khả năng tích hợp mạnh mẽ với nhiều ngôn ngữ lập trình khác. Khả năng này làm cho Python trở thành lựa chọn phổ biến cho các dự án yêu cầu sự kết hợp giữa các thành phần phần mềm được phát triển bằng các ngôn ngữ khác. Điều này mở ra nhiều khả năng kết hợp giữa các thư viện và

frameworks khác nhau, tận dụng sức mạnh của Python trong môi trường lập trình đa ngôn ngữ.

Thư Viện và Framework Mạnh Mẽ:

+ Python có cộng đồng phát triển mạnh mẽ và đa dạng, mang lại nhiều thư viện và frameworks mạnh mẽ hỗ trợ lập trình mạng. Điều này giúp giảm thời gian và công sức cần thiết cho việc phát triển ứng dụng mạng. Các thư viện như Socket, Requests, và frameworks như Twisted, Flask, Django cung cấp các công cụ tiện ích để xây dựng và triển khai các ứng dụng mạng đa dạng và hiệu quả.

Quản Lý Bộ Nhớ Tự Động:

+ Python có bộ thu gom rác tự động, làm giảm gánh nặng của nhà phát triển trong việc quản lý bộ nhớ. Cơ chế này tự động thu hồi bộ nhớ không còn sử dụng, giúp người phát triển tránh được các lỗi liên quan đến quản lý bộ nhớ như rò rỉ bộ nhớ. Điều này cung cấp sự thuận tiện và giảm khả năng xuất hiện lỗi do quản lý bộ nhớ không đúng.

=> Những đặc điểm trên làm cho Python trở thành một ngôn ngữ lập trình linh hoạt và hiệu quả, đặc biệt phù hợp trong lĩnh vực lập trình mạng, nơi sự dễ đọc, đơn giản và khả năng tích hợp linh hoạt rất quan trọng.

**Nhược điểm của Python**

Tốc Độ Thực Thi:

+ Python thường chậm hơn so với các ngôn ngữ biên dịch như C++ hoặc Java, đặc biệt trong các ứng dụng yêu cầu xử lý đa luồng lớn.

Khả Năng Mở Rộng:

+ Khó khăn khi xử lý các ứng dụng mạng lớn và phức tạp, đặc biệt là khi cần tối ưu hóa hiệu suất.

### Các công cụ hỗ trợ cho lập trình mạng trong Python

### Socket

*Chức Năng:*

+ Module socket cung cấp giao diện cho việc phát triển ứng dụng mạng trong Python.

+ Hỗ trợ tạo, quản lý và tương tác với các kết nối mạng, bao gồm cả TCP và

UDP.

Sử Dụng Thông Thường:

+ Sử dụng để xây dựng các ứng dụng mạng như máy chủ-client, truyền file qua mạng, và các ứng dụng mạng khác.

### Threading

Chức Năng:

+ Module threading hỗ trợ lập trình đa luồng trong Python.

+ Cho phép tạo và quản lý nhiều luồng thực thi cùng một lúc để tận dụng tối đa khả năng xử lý của hệ thống.

Sử Dụng Thông Thường:

+ Sử dụng để tăng hiệu suất ứng dụng bằng cách thực hiện các tác vụ đồng thời, như xử lý nền, cập nhật giao diện người dùng, và các tác vụ khác.

### Os

Chức Năng:

+ Module os cung cấp các hàm để tương tác với hệ điều hành, cung cấp khả năng thực hiện các thao tác hệ thống.

+ Cho phép thực hiện các tác vụ như thay đổi thư mục làm việc, tạo thư mục, thực thi các lệnh hệ thống, và quản lý tệp tin.

Sử Dụng Thông Thường:

+ Sử dụng để tương tác với hệ thống file, thực hiện các tác vụ hệ thống, và kiểm soát môi trường thực thi.

### PyQt5

Chức Năng:

+ PyQt5 là một wrapper của toolkit đồ họa Qt cho Python.

+ Cung cấp các công cụ để xây dựng giao diện người dùng đồ họa và ứng dụng đa nền tảng.

Sử Dụng Thông Thường:

+ Sử dụng để phát triển ứng dụng đồ họa, ứng dụng desktop, và các ứng dụng có giao diện người dùng phức tạp.

### Kết luận

Python không chỉ là ngôn ngữ lập trình mạng mạnh mẽ mà còn là sự chọn lựa lý tưởng cho việc phát triển các ứng dụng chat. Với sự linh hoạt, độ đơn giản và sự hỗ trợ mạnh mẽ từ các thư viện và frameworks, Python đặc biệt thích hợp cho việc xây dựng các ứng dụng mạng đa người dùng hiệu quả và dễ bảo trì.

Việc sử dụng Python trong lập trình mạng và xây dựng chương trình chat không chỉ mang lại sự thuận tiện trong quá trình phát triển mà còn đảm bảo hiệu suất và độ ổn định của ứng dụng.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Phân tích yêu cầu

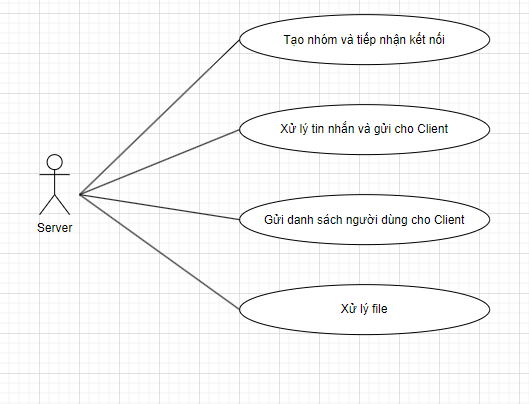
Sử dụng giao thức TCP/IP và mô hình Client – Server để xây dựng chương trình chat bằng ngôn ngữ Python.

* + - Sử dụng giao thức TCP/IP và mô hình Client – Server để thiết lập kết nối giữa máy chủ và máy khách
    - Cung cấp giao diện đơn giản để người dùng dễ dàng tương tác với hệ thống
    - Server có khả năng xử lý nhiều kết nối từ các client khác nhau đồng thời, giúp các người dùng có thể tham gia vào cuộc trò chuyện cùng một lúc.
    - Cung cấp các chức năng cơ bản của một ứng dụng chat như tham gia phòng chat, gửi và nhận tin nhắn chung, gửi và nhận tin nhắn riêng, gửi file, gửi icon.

## Phân tích thiết kế ứng dụng:

### Phía Server

### Biểu đồ usecase tổng quát



*Hình : Usecase tổng quát Server*

Actor Server: Phía Server có thể tạo phòng chat và tiếp nhận kết nối từ các Client, xử lý tin nhắn và icon để gửi chúng đến các Client, quản lý danh sách người dùng trong phòng chat, xử lý file.

### Đặc tả chức năng

* + - * + Tạo nhóm và tiếp nhận kết nối:

Server khởi chạy ứng dụng để tạo phòng chờ với IP và PORT Server lắng nghe và tiếp nhận kết nối từ Client

Thêm Client vào danh sách người dùng trong phòng

* + - * + Xử lý tin nhắn và gửi cho Client:

Server tiếp nhận thông tin từ Client, kiểm tra người gửi, người nhận và nội dung của tin nhắn

Sau khi kiểm tra thì Server sẽ chuyển tin nhắn đến Client theo đúng yêu cầu

* + - * + Gửi danh sách người dùng cho Client:

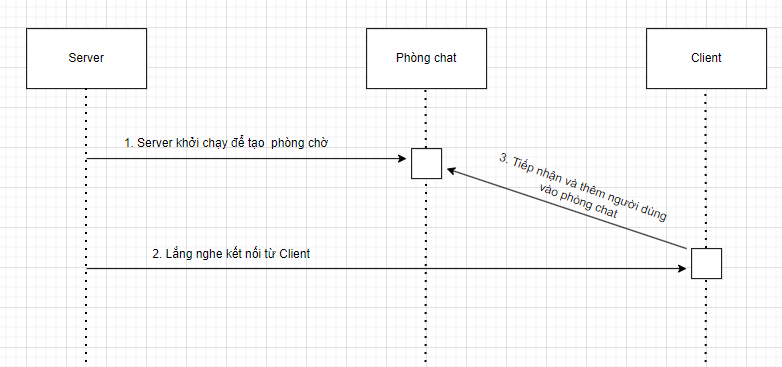
Khi có người dùng kết nối với phòng thì Server sẽ tạo một danh sách người dùng và gửi danh sách đó đến cho Client

* + - * + Xử lý file:

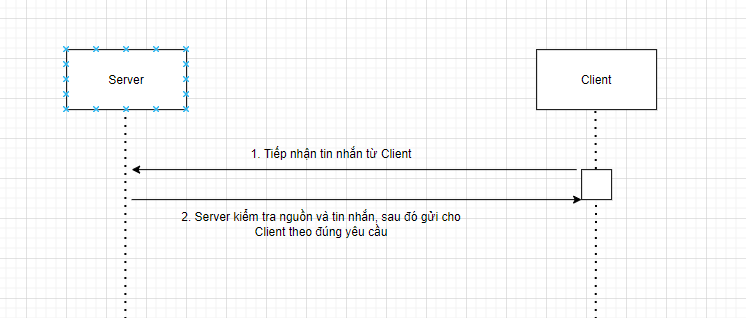
Server tiếp nhận thông tin là file từ Client, kiểm tra nguồn và thông tin sau đó gửi nó cho Client

### Biểu đồ trình tự

* + - * + Tạo nhóm và tiếp nhận kết nối

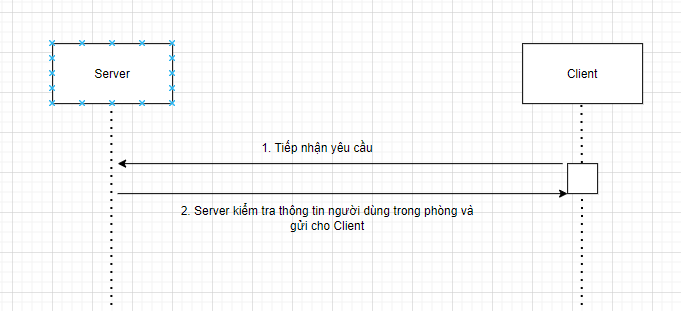


*Hình : Tạo nhóm và tiếp nhận kết nối*

* + - * + Xử lý tin nhắn và gửi cho Client

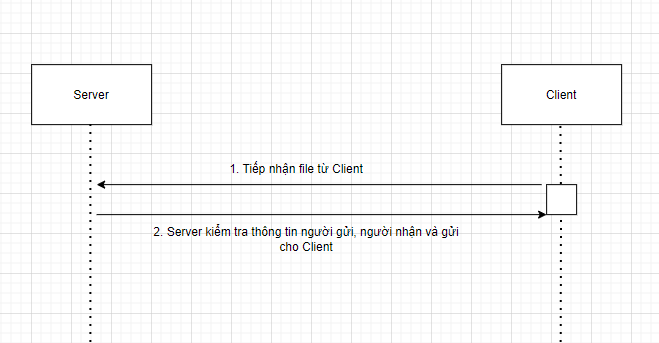
***Hình : Xử lý tin nhắn và gửi cho Client***

* + - * + Gửi danh sách người dùng cho Client



***Hình : Gửi danh sách người dùng cho Client***

* + - * + Xử lý file



***Hình : Xử lí file***

### Phía Client

### Biểu đồ usecase tổng quát

***Hình : Usecase tổng quát Client***

Actor Client: Người dùng có thể nhập thông tin và tham gia phòng chat, gửi tin nhắn chung và riêng, gửi file, hiển thị danh sách người dùng và gửi icon.

### Đặc tả chức năng

* + - * + Nhập thông tin và tham gia nhóm chat Người dùng khởi chạy ứng dụng phía Client

Nhập thông tin và kết nối với phòng chat thông qua dịa chỉ IP và PORT cụ thể

* + - * + Gửi tin nhắn chung

Người dùng nhập tin nhắn vào ô văn bản Lựa chọn gửi cho tất cả mọi người

* + - * + Gửi tin nhắn riêng

Người dùng nhập tin nhắn vào ô văn bản Lựa chọn người muốn gửi tin nhắn riêng

* + - * + Gửi file

Người dùng chọn tệp tin từ thiết bị.

Gửi tệp tin đến tất cả người dùng trong phòng chat.

* + - * + Hiển thị danh sách người dùng

Hiển thị danh sách người dùng đang tham gia trong phòng chat.

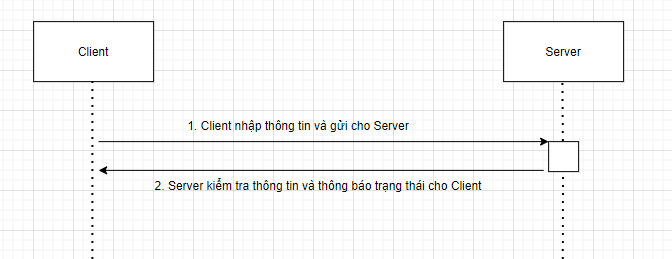
Cập nhật danh sách người dùng khi có sự thay đổi (người mới tham gia hoặc rời khỏi phòng).

* + - * + Gửi icon

Người dùng chọn biểu tượng icon muốn gửi Lựa chọn để gửi tintin nhắn vào nhóm chat

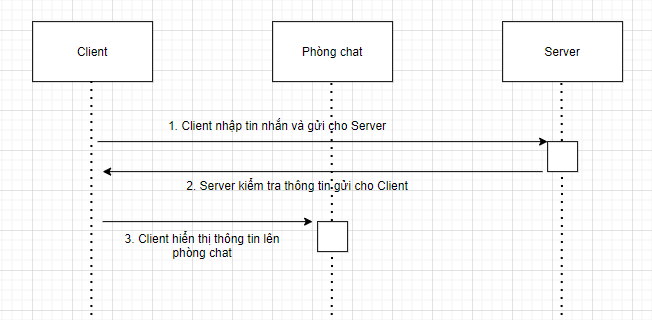
### Biểu đồ trình tự

* + - * + Nhập thông tin và tham gia nhóm chat



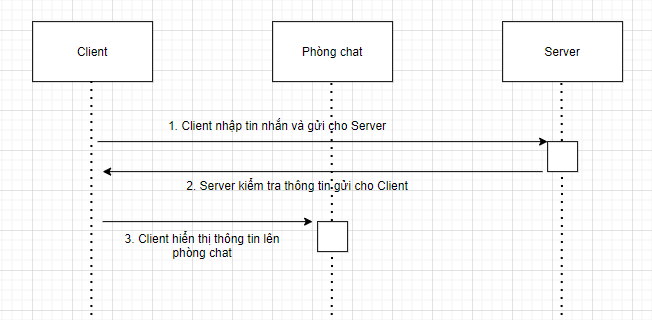
*Hình 9: Nhập thông tin và tham gia nhóm chat*

* + - * + Gửi tin nhắn chung



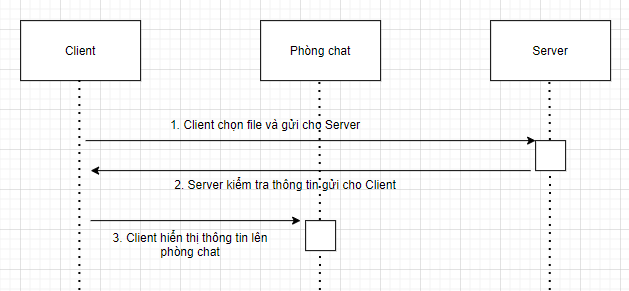
*Hình 10: Gửi tin nhắn chung*

* + - * + Gửi tin nhắn riêng



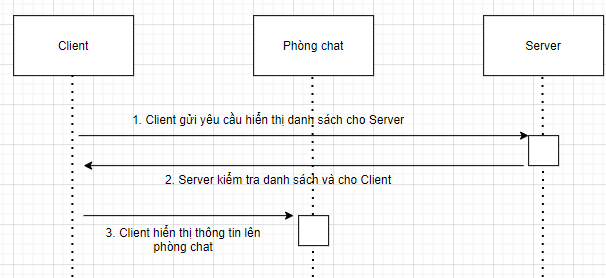
*Hình : Gửi tin nhắn riêng*

* + - * + Gửi file



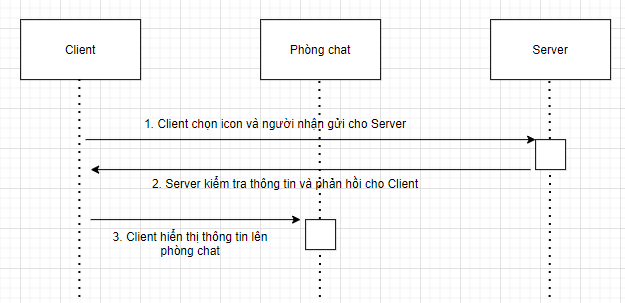
*Hình : Gửi file*

* + - * + Hiển thị danh sách người dùng



*Hình : Hiển thị danh sách người dùng*

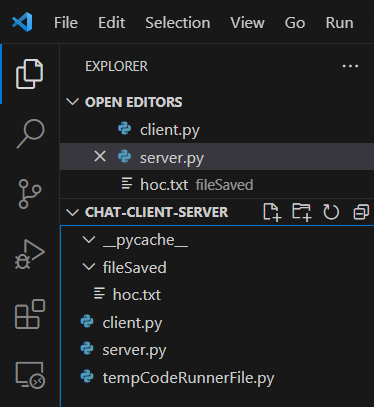
* + - * + Gửi icon



*Hình : Gửi icon*

# KẾT QUẢ CHƯƠNG TRÌNH

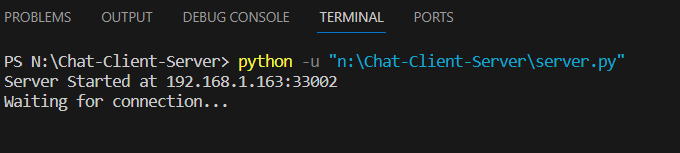
## Cấu trúc thư mục



HÌnh : Cấu trúc thư mục

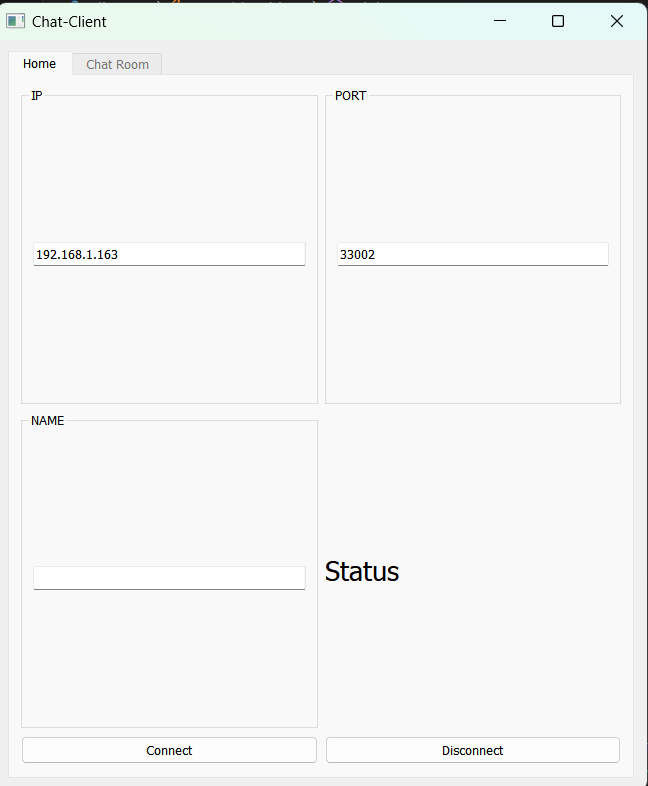
## Khởi chạy Server

* + - * + Đầu tiên, chúng ta khởi chạy Server để lắng nghe các kết nối từ Client.

****

***Hình : Khởi chạy Server***

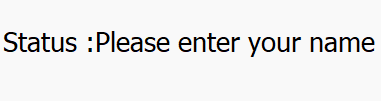
## Khởi chạy Client

****

*Hình : Khởi chạy Client*

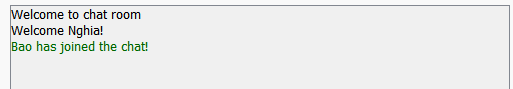
Sau khi khởi chạy Client, chúng ta điền các thông tin cơ bản như địa chỉ IP, PORT và NAME để gửi kết nối tới Server.

Nếu không điền đủ thông tin hoặc sai thông tin thì sẽ thông báo đến Client.



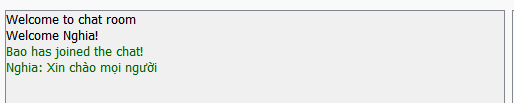
***Hình : Thông báo trạng thái***

* + - * + Sau khi điền đầy đủ thông tin chính xác và ấn Connect thì sẽ mở ra giao diện phòng chat.



***Hình : Thông báo người dùng mới***

Khi có tin nhắn mới thì sẽ hiển thị vào hộp thoại dưới dạng: Name: Message. Trong đó Name là tên người dùng gửi tin nhắn và Message là thông tin.



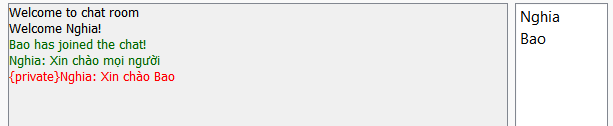
***Hình : Hộp thoại tin nhắn mới***

Người dùng có thể chọn một người dùng khác trong phòng chat để gửi tin nhắn riêng tư.



***Hình : Lựa chọn người nhận***

Tin nhắn riêng tư chỉ hiển thị đối với người gửi và người được chọn để gửi, những người khác trong phòng chat sẽ không thấy được tin nhắn đó. Tin nhắn được hiển thị dưới dạng: {private}Name: Message trong đó {private} thể hiện đây là tin nhắn riêng tư và màu của tin nhắn sẽ chuyển thành màu đỏ.



***Hình : Hộp thoại tin nhắn riêng tư***

Người dùng có thể click vào nút Choose File để chọn File gửi đi. Sau khi chọn File và gửi thành công thì sẽ thông báo lên hộp thoại tin nhắn. Tin nhắn hiển thị dưới dạng: FileName: received a file.



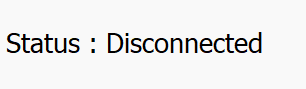
***Hình : Thông báo nhận file***

Người dùng có thể gửi nhanh các icon bằng cách click vào icon muốn gửi đi.



***Hình : Hộp thoại tin nhắn icon***

Khi người dùng muốn rời khỏi phòng chat, người dùng có thể quay lại giao diện ban đầu, chọn vào Disconnected thì sẽ lập tức rời khỏi phòng chat.



***Hình : Thông báo trạng thái rời khỏi nhóm***

Và khi có người dùng rời khỏi phòng chat thì sẽ thông báo tới các người dùng còn lại trong phòng chat.

KẾT LUẬN

* Nhìn chung, sau khi kết thúc đồ án em đã đạt được những điều sau:

+ Nắm bắt được các giao thức trong kết nối mạng .

+ Hoàn thành quá trình minh họa trong chương trình.

+ Ứng dụng đã hoàn thiện đầy đủ các chức năng cơ bản.

* Bên cạnh đó, còn có một số hạn chế như sau:

+ Giao diện của nhóm còn chưa được đẹp

+ Trong quá trình sử dụng có thể gặp một số lỗi hiển thị.

+ Chưa áp dụng được trí tuệ nhân tạo trong quá trình kiểm duyệt nội dung bài viết mà cần đến sự can thiệp thủ công của admin.

**Kết quả chưa đạt được**

* Chưa hoàn thiện chương trình minh họa: Có thể chương trình minh họa chưa hoàn thiện hoặc có các lỗi trong việc thực hiện quá trình tạm dừng và tiếp tục bằng tín hiệu. Điều này có thể dẫn đến việc không đạt được mục tiêu của đồ án.
* Thiếu tính ứng dụng thực tế: Nếu chương trình minh họa chỉ là một ví dụ lý thuyết mà không áp dụng vào thực tế hoặc không cung cấp giá trị thực tế cho người sử dụng, đây cũng là một hạn chế trong đồ án.
* Hiệu suất yếu hoặc không ổn định: Nếu chương trình có hiệu suất yếu hoặc gặp vấn đề về sự ổn định khi thực hiện quá trình tạm dừng và tiếp tục, điều này có thể dẫn đến kết quả chưa đạt được.

**Hướng phát triển**

* Mở rộng tính năng: Bổ sung tính năng mới vào chương trình để tạo ra một ứng dụng đa dạng và phong phú hơn. Các tính năng mới có thể bao gồm quản lý tài nguyên, đa luồng (multithreading), giao tiếp mạng, và quản lý lỗi.
* Cải thiện hiệu suất: Tối ưu hóa hiệu suất chương trình để đảm bảo rằng nó hoạt động mượt mà và nhanh chóng, đặc biệt khi xử lý tín hiệu và dữ liệu lớn. Sử dụng các công cụ và kỹ thuật tối ưu hóa mã nguồn để giảm tiêu tốn tài nguyên.
* Tương tác giao diện: Cải thiện giao diện người dùng để làm cho ứng dụng dễ sử dụng và thân thiện hơn với người dùng cuối. Thêm các tính năng như biểu đồ, báo cáo, và mô tả chi tiết cho quá trình tạm dừng và tiếp tục.
* Tích hợp với hệ thống thực tế: Tạo tích hợp với các hệ thống hoặc ứng dụng thực tế để sử dụng quá trình tạm dừng và tiếp tục trong môi trường thực tế. Điều này có thể làm cho ứng dụng trở thành một công cụ hữu ích cho các dự án thực tế.

**Bài học kinh nghiệm**

* Lập kế hoạch cẩn thận: Kế hoạch trước khi bắt đầu dự án là rất quan trọng. Điều này bao gồm việc xác định mục tiêu, tính năng cần phát triển và thời gian dự kiến cho dự án. Kế hoạch này giúp bạn theo dõi tiến độ và đảm bảo rằng dự án được hoàn thành đúng hạn.
* Tối ưu hóa hiệu suất: Học hỏi về việc tối ưu hóa hiệu suất của chương trình là quan trọng để đảm bảo rằng ứng dụng hoạt động mượt mà và hiệu quả. Sử dụng các kỹ thuật tối ưu hóa và công cụ phân tích để giảm tiêu tốn tài nguyên và tăng hiệu suất.
* Thử nghiệm và kiểm tra: Thực hiện kiểm tra tự động và kiểm tra thủ công để đảm bảo rằng chương trình hoạt động đúng đắn và ổn định. Kiểm tra cẩn thận có thể giúp phát hiện và sửa lỗi trước khi chương trình được triển khai.

TÀI LIỆU THAM KHẢO

|  |  |  |
| --- | --- | --- |
| [1] |  | Advanced Programming in the UNIX® Environment Third Edition – |
| [2] |  | W.Richard Stevens Stephen A. Rago |
| [3] |  | website Ftech.ai : <https://vietjack.com/> |
| [4] |  | Giáo trình Lập trình mạng: https://nst.utb.edu.vn/attachments/article/205/lap\_trinh\_mang.pdf |
|  |  |  |