|  |  |
| --- | --- |
| BÁO CÁO Mô hình Client-Server và Sockets  Bản Báo Cáo | Tóm tắt  Mô hình Client-Server:Máy chủ cung cấp dịch vụ, máy khách yêu cầu.Dễ mở rộng, an toàn, hiệu quả, nhưng phụ thuộc và tốn kém.Sockets:Giao diện cho phép truyền thông qua mạng.Có TCP (tin cậy) và UDP (không tin cậy).Ứng dụng:Web, email, chat, chia sẻ tập tin, game,...Kết luận:Công cụ mạnh mẽ để xây dựng ứng dụng mạng.  Vũ Thành Trung  Lập Trình Mạng |

Mục lục

[**Lời mở đầu** 3](#_Toc163506977)

[**1. Giới thiệu:** 4](#_Toc163506978)

[**2. Sockets:** 4](#_Toc163506979)

[**2.1 Định nghĩa:** 4](#_Toc163506980)

[**2.2 Cách thức hoạt động:** 4](#_Toc163506981)

[**2.3 Các loại sockets:** 4](#_Toc163506982)

[**3. Mô hình Client-Server:** 4](#_Toc163506983)

[**3.1 Cách thức hoạt động:** 4](#_Toc163506984)

[**3.2 Ưu và nhược điểm:** 4](#_Toc163506985)

[**4. Ví dụ minh họa:** 4](#_Toc163506986)

[**4.1 Thiết lập kết nối socket:** 4](#_Toc163506987)

[**4.2 Gửi và nhận dữ liệu:** 6](#_Toc163506988)

[**5. Phân tích và đánh giá:** 7](#_Toc163506989)

[**5.1 Ưu điểm của mô hình Client-Server và sockets:** 7](#_Toc163506990)

[**5.2 Nhược điểm của mô hình Client-Server và sockets:** 7](#_Toc163506991)

[**5.3 Ứng dụng thực tế:** 7](#_Toc163506992)

[**6. Kết luận:** 8](#_Toc163506993)

# **Lời mở đầu**

Mô hình Client-Server và sockets là hai khái niệm cơ bản trong lập trình mạng. Mô hình Client-Server phân chia các ứng dụng mạng thành hai thành phần: máy chủ (server) và máy khách (client). Sockets là giao diện lập trình ứng dụng (API) cho phép các chương trình truyền thông tin qua mạng.

Báo cáo này cung cấp một cái nhìn tổng quan về mô hình Client-Server và sockets. Báo cáo bao gồm các nội dung sau:

* **Giới thiệu:** Giới thiệu về mô hình Client-Server và sockets.
* **Sockets:** Định nghĩa, cách thức hoạt động, và các loại sockets.
* **Mô hình Client-Server:** Cách thức hoạt động, ưu và nhược điểm.
* **Ví dụ minh họa:** Ví dụ về cách thiết lập kết nối socket và gửi/nhận dữ liệu.
* **Phân tích và đánh giá:** Ưu và nhược điểm của mô hình Client-Server và sockets, ứng dụng thực tế.
* **Kết luận:** Tóm tắt nội dung báo cáo.

Báo cáo này hướng đến các đối tượng sau:

* Sinh viên ngành khoa học máy tính hoặc công nghệ thông tin.
* Lập trình viên muốn tìm hiểu về lập trình mạng.
* Bất kỳ ai quan tâm đến mô hình Client-Server và sockets.

**Hy vọng báo cáo này sẽ cung cấp cho bạn những thông tin hữu ích về mô hình Client-Server và sockets.**

# **1. Giới thiệu:**

Mô hình Client-Server và sockets là hai khái niệm cơ bản trong lập trình mạng. Mô hình Client-Server phân chia các ứng dụng mạng thành hai thành phần: máy chủ (server) và máy khách (client). Sockets là giao diện lập trình ứng dụng (API) cho phép các chương trình truyền thông tin qua mạng.

# **2. Sockets:**

## **2.1 Định nghĩa:**

Socket là điểm cuối của một kết nối mạng hai chiều. Nó có thể được ví như một ổ cắm điện, nơi các chương trình có thể kết nối với nhau để trao đổi dữ liệu.

## **2.2 Cách thức hoạt động:**

Sockets hoạt động dựa trên mô hình giao tiếp giữa các tiến trình (IPC). Khi hai chương trình muốn giao tiếp với nhau, chúng sẽ tạo ra một socket và kết nối với nhau thông qua socket đó. Dữ liệu được truyền qua socket dưới dạng các gói tin.

## **2.3 Các loại sockets:**

Có hai loại sockets chính:

* **TCP sockets:** Cung cấp kết nối tin cậy, đảm bảo dữ liệu được truyền đi đầy đủ và chính xác.
* **UDP sockets:** Cung cấp kết nối không tin cậy, không đảm bảo dữ liệu được truyền đi đầy đủ và chính xác.

# **3. Mô hình Client-Server:**

## **3.1 Cách thức hoạt động:**

Trong mô hình Client-Server, máy chủ đóng vai trò là trung tâm cung cấp dịch vụ cho các máy khách. Các máy khách gửi yêu cầu đến máy chủ và máy chủ sẽ xử lý yêu cầu và gửi trả kết quả cho máy khách.

## **3.2 Ưu và nhược điểm:**

**Ưu điểm:**

* Dễ dàng mở rộng: Máy chủ có thể xử lý nhiều yêu cầu từ nhiều máy khách.
* An toàn: Máy chủ có thể kiểm soát quyền truy cập vào dữ liệu.
* Hiệu quả: Máy chủ có thể tối ưu hóa việc sử dụng tài nguyên.

**Nhược điểm:**

* Phụ thuộc vào máy chủ: Nếu máy chủ gặp sự cố, tất cả các máy khách sẽ bị ảnh hưởng.
* Chi phí cao: Máy chủ cần có cấu hình mạnh để xử lý nhiều yêu cầu.

# **4. Ví dụ minh họa:**

## **4.1 Thiết lập kết nối socket:**

Đoạn mã

// Máy chủ

using System;

using System.Net;

using System.Net.Sockets;

class Server

{

static void Main(string[] args)

{

// Tạo socket

Socket serverSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

// Gán địa chỉ và cổng cho socket

serverSocket.Bind(new IPEndPoint(IPAddress.Any, 8000));

// Bắt đầu lắng nghe kết nối

serverSocket.Listen(10);

// Chấp nhận kết nối từ máy khách

Socket clientSocket = serverSocket.Accept();

// ...

// Đóng socket

serverSocket.Close();

clientSocket.Close();

}

}

// Máy khách

using System;

using System.Net;

using System.Net.Sockets;

class Client

{

static void Main(string[] args)

{

// Tạo socket

Socket clientSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

// Kết nối đến máy chủ

clientSocket.Connect(new IPEndPoint(IPAddress.Parse("127.0.0.1"), 8000));

// ...

// Đóng socket

clientSocket.Close();

}

}

## **4.2 Gửi và nhận dữ liệu:**

Đoạn mã

// Máy chủ

byte[] data = new byte[1024];

int receivedBytes = clientSocket.Receive(data);

// Xử lý dữ liệu nhận được

clientSocket.Send(data, receivedBytes, SocketFlags.None);

// Máy khách

byte[] data = new byte[1024];

int receivedBytes = serverSocket.Receive(data);

// Xử lý dữ liệu nhận được

serverSocket.Send(data, receivedBytes, SocketFlags.None);

# **5. Phân tích và đánh giá:**

## **5.1 Ưu điểm của mô hình Client-Server và sockets:**

* Dễ dàng sử dụng và triển khai.
* Khả năng mở rộng cao.
* An toàn và bảo mật.
* Hiệu quả và tiết kiệm tài nguyên.

## **5.2 Nhược điểm của mô hình Client-Server và sockets:**

* Phụ thuộc vào máy chủ.
* Chi phí cao cho máy chủ.
* Khó khăn trong việc gỡ lỗi.

## **5.3 Ứng dụng thực tế:**

**Mô hình Client-Server và sockets được sử dụng trong nhiều ứng dụng mạng, bao gồm:**

* **Web:** Máy chủ web cung cấp nội dung web cho các trình duyệt web. Khi bạn truy cập một trang web, trình duyệt web của bạn sẽ gửi một yêu cầu đến máy chủ web. Máy chủ web sẽ xử lý yêu cầu và gửi lại nội dung trang web cho trình duyệt web của bạn.
* **Email:** Máy chủ email lưu trữ và chuyển tiếp email. Khi bạn gửi một email, email của bạn sẽ được gửi đến máy chủ email của người nhận. Máy chủ email của người nhận sẽ lưu trữ email cho đến khi người nhận truy cập vào hộp thư của họ.
* **Chat:** Máy chủ chat cho phép người dùng trò chuyện trực tuyến. Khi bạn gửi một tin nhắn chat, tin nhắn của bạn sẽ được gửi đến máy chủ chat. Máy chủ chat sẽ chuyển tiếp tin nhắn đến tất cả các người dùng khác đang tham gia vào cuộc trò chuyện.
* **File sharing:** Máy chủ chia sẻ tập tin cho phép người dùng chia sẻ tập tin với nhau. Khi bạn chia sẻ một tập tin, tập tin của bạn sẽ được tải lên máy chủ chia sẻ tập tin. Người dùng khác có thể tải xuống tập tin từ máy chủ chia sẻ tập tin.
* **Games:** Máy chủ game cho phép người chơi chơi game trực tuyến. Khi bạn chơi một game trực tuyến, máy tính của bạn sẽ kết nối với máy chủ game. Máy chủ game sẽ xử lý tất cả các logic của game và gửi lại cập nhật cho máy tính của bạn.

**Ngoài ra, mô hình Client-Server và sockets còn được sử dụng trong nhiều ứng dụng khác, bao gồm:**

* **Hệ thống quản lý cơ sở dữ liệu:** Máy chủ cơ sở dữ liệu lưu trữ và truy cập dữ liệu. Các ứng dụng khách truy cập dữ liệu từ máy chủ cơ sở dữ liệu.
* **Hệ thống giám sát:** Máy chủ giám sát thu thập dữ liệu từ các thiết bị khác nhau. Các ứng dụng khách hiển thị dữ liệu được thu thập bởi máy chủ giám sát.
* **Hệ thống điều khiển tự động:** Máy chủ điều khiển tự động điều khiển các thiết bị khác nhau. Các ứng dụng khách giám sát và điều khiển các thiết bị được điều khiển bởi máy chủ điều khiển tự động.

**Mô hình Client-Server và sockets là những công cụ mạnh mẽ có thể được sử dụng để xây dựng nhiều loại ứng dụng mạng.**

# **6. Kết luận:**

Mô hình Client-Server và sockets là những công cụ quan trọng trong lập trình mạng. Mô hình Client-Server cung cấp một cách thức hiệu quả để tổ chức các ứng dụng mạng, và sockets cung cấp một giao diện lập trình đơn giản để truyền thông qua mạng.

**Ứng dụng thực tế:**

Mô hình Client-Server và sockets được sử dụng trong nhiều ứng dụng mạng, bao gồm:

* **Web:** Máy chủ web cung cấp nội dung web cho các trình duyệt web.
* **Email:** Máy chủ email lưu trữ và chuyển tiếp email.
* **Chat:** Máy chủ chat cho phép người dùng trò chuyện trực tuyến.
* **File sharing:** Máy chủ chia sẻ tập tin cho phép người dùng chia sẻ tập tin với nhau.
* **Games:** Máy chủ game cho phép người chơi chơi game trực tuyến.