Trường Đại học Bách Khoa Hà Nội

Viện Công Nghệ Thông Tin và Truyền Thông

──────── \* ───────



BÁO CÁO TÍNH TOÁN PHÂN TÁN

**Đề tài: Ứng dụng phân tán cho phép lưu trữ và gửi nhận tệp dữ liệu**

*Giảng viên hướng dẫn*: **TS. Nguyễn Hữu Đức**

*Sinh viên thực hiện*: Nguyễn Văn Tuân – 20144838

Nguyễn Văn Ngọc – 20143223

***Hà Nội, 5/2018***

1

[**PHẦN 1. LỜI NÓI ĐẦU** 3](#_Toc514189634)

[**PHẦN 2 : GIỚI THIỆU BÀI TOÁN** 4](#_Toc514189635)

[**1.** **Giới thiệu bài toán** 4](#_Toc514189636)

[**2.** **Phân tích bài toán** 8](#_Toc514189637)

[**3.** **Kết quả đạt được và hướng phát triển** 11](#_Toc514189638)

[**PHẦN 3 : CÀI ĐẶT VÀ DEMO CHƯƠNG TRÌNH** 11](#_Toc514189639)

[**PHẦN 4 : TỔNG KẾT** 16](#_Toc514189640)

[**1.** **Danh sách tài liệu tham khảo** 16](#_Toc514189641)

# **PHẦN 1. LỜI NÓI ĐẦU**

*Trong những năm qua, cùng với sự phát triển như vũ bão của khoa học kĩ thuật, ngành công nghệ thông tin cũng biến đổi không ngừng để đáp ứng vớimọi nhu cầu mới của con người. Trong số những phát minh vĩ đại nhất của thế kỉ trước, mạng máy tính ( Computer Network) là một hệ thống đem lại nhiều lợi ích to lớn nhất cho nhân loại. Mạng máy tính ngày nay cho phép hàng tỉ máy tính trên trái đất đã và đang kết nối và làm việc việc với nhau với tốc độ cao( từ vài chục đến hàng nghìn kbps)*

*Điều kì diệu trên đã trở thành hiện thực và ngày càng phát triển mạnh mẽ hơn nữa chỉ khi có một nhân tố quan trọng làm nền tảng cơ sở lí thuyết chochúng đã được nghiên cứu. Đó chính là những lí thuyết nghiên cứu liên quan đến tính toán phân tán. Nhờ có lí thuyết hệ phân tán, mà các vi xử lí đơn lẻ - nằm trên các máy tính đơn lẻ (kèm theo là các thiết bị ngoại vi và các thiết bị đầu cuốilàm việc từ xa)- mà chúng vẫn thường được gọi là các hệ thống tập trung(Centralized Systems) đã được kết nối lại, làm việc trên nhiều giao thức, nhiềukiểu kết nối và trên nhiều lãnh thổ khác nhau. Và khi các hệ thống tập trung đóđã được kết nối lại theo nhiều nhóm, nhiều cách thức khác nhau và làm việc dựa trên những nguyên lí cơ bản của hệ phân tán thì chúng sẽ được gọi là các hệ thống phân tán (Distributed Systems), điển hình nhất của các hệ phân tán chính là mạng máy tính mà chúng đa đang sử dụng ngày nay.*

*Dưới đây là phần trình bày của nhóm em về hệ thống chia sẻ lưu trữ và gửi nhận tệp tin dữ liệu mà chúng em đã xây dựng được. Chúng em rất mong nhận được sự góp ý nhiệt tình từ thầy để hoàn thiện thêm hệ thống và cũng hi vọng hệ thống này sẽ giúp ích cho mọi người.*

*Nhóm sinh viên thực hiện*

# **PHẦN 2 : GIỚI THIỆU BÀI TOÁN**

### **Giới thiệu bài toán**

#### **Mô tả ứng dụng**

Ứng dụng phân tán đơn giản cho phép lưu trữ và gửi nhận tệp dữ liệu. Yêu cầu:

* Dữ liệu được phía client được gửi lên (upload) máy Server, và ngược lại client có thể tải (download) dữ liệu từ máy server về.
* Cơ chế upload và download dữ liệu được thực hiện thông qua cơ chế phân mảnh.
* Dữ liệu được chia thành các part nhỏ và tải lên hoặc download về các mảnh này, sau khi quá trình upload và download các mảnh hoàn tất, các mảnh được ghép lại với nhau thành file hoàn chỉnh.
* Cung cấp giao diện đơn giản (dạng Explorer) để hiển thị các chức năng, download, upload, dừng upload, download, tiếp tục upload, download (khi chưa hoàn thành).
* Xử lý tương tranh dữ liệu giữa 2 các lần upload lên Server và download về Client.

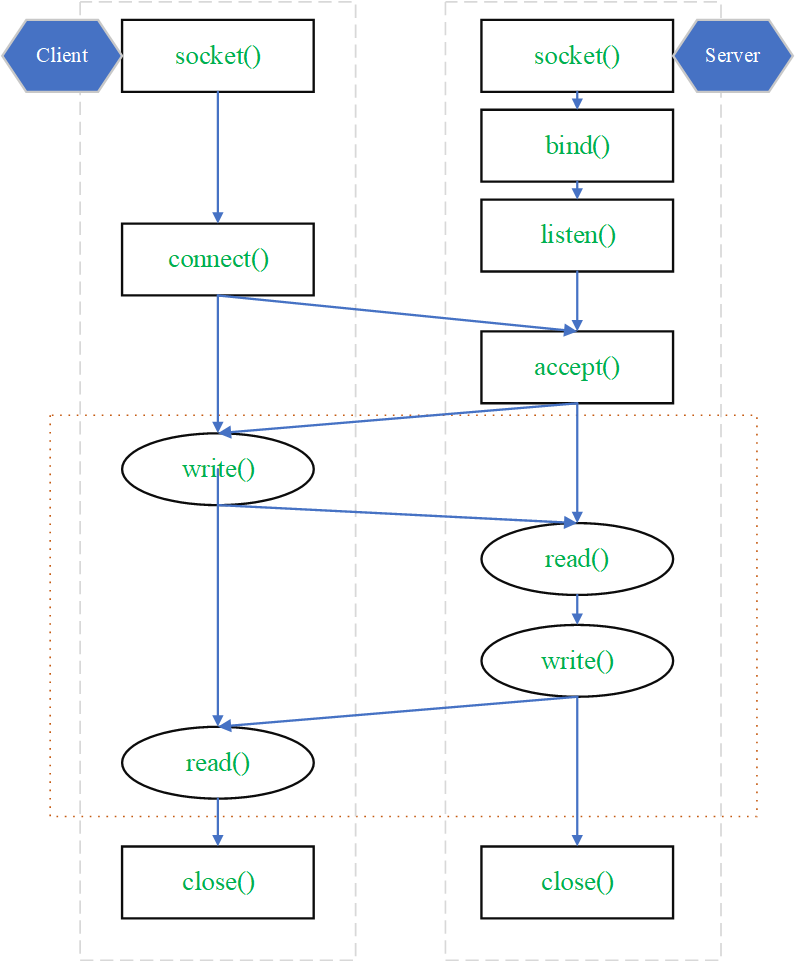
#### **Mô hình phân tán**

* + Ứng dụng sử dụng mô hình Client – Server sử dụng socket ở chế độ kết nối TCP
  + Socket là phương tiện hiệu quả để xây dựng các ứng dụng theo kiến trúc Client-Server.
  + Socket là thiết bị truyền thông hai chiều gửi và nhận dữ liệu từ máy khác

##### **1.2.1. Đặc điểm của Socket :**

* Có một đường kết nối (địa chỉ IP) giữa 2 tiến trình.
* Một trong 2 tiến trình kia phải đợi tiến trình kia yêu cầu kết nối.
* Có thể dùng để liên lạc theo mô hình client và sever
* Mô hình client /sever thì sever lắng nghe và chấp nhận từ client
* Mỗi thông điệp gửi phải có xác nhận trả về
* Các gói tin chuyển đi tuần tự.
  + Để có thể thực hiện các cuộc giao tiếp , một trong 2 quá trình phải công bố số hiệu cổng của socket mà mình đang sử dụng
  + Mỗi công giao tiếp phải thể hiện một địa chỉ xác định trong hệ thống.
  + Khi quá trình được gán một số hiệu cổng , nó có thể nhận dữ liệu gửi đến chỗ này từ quá trình khác

##### **1.2.2. Mô hình Client – Server sử dụng socket ở chế độ TCP**



Hình 1 : Mô hình Client – Server

**Các giai đoạn của mô hình Client – Server sử dụng Socket ở chế độ TCP**

* **Giai đoạn 1 :**
* Server tạo Socket, gán số hiệu cổng và lắng nghe yêu cầu nối kết. Server sẵn sàng phục vụ Client.socket(): Server yêu cầu tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển.
* bind(): Server yêu cầu gán số hiệu cổng (port) cho socket.
* listen(): Server lắng nghe các yêu cầu nối kết từ các client trên cổng đã được gán.
* **Giai đoạn 2 :**
* socket(): Client yêu cầu tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển, thông thường hệ thống tự động gán một số hiệu cổng còn rảnh cho socket của Client.
* connect(): Client gởi yêu cầu nối kết đến server có địa chỉ IP và Port xác định.
* accept(): Server chấp nhận nối kết của client, khi đó một kênh giao tiếp ảo được hình thành, Client và server có thể trao đổi thông tin với nhau thông qua kênh ảo này.
* **Giai đoạn 3 :** Trao đổi thông tin giữa Client và Server.
* Sau khi chấp nhận yêu cầu nối kết, thông thường server thực hiện lệnh read() và nghẽn cho đến khi có thông điệp yêu cầu (Request Message) từ client gởi đến.
* Server phân tích và thực thi yêu cầu. Kết quả sẽ được gởi về client bằng lệnh write().
* Sau khi gởi yêu cầu bằng lệnh write(), client chờ nhận thông điệp kết quả (ReplyMessage) từ server bằng lệnh read().
* **Giai đoạn 4 :** Kết thúc phiên làm việc.
* Các câu lệnh read(), write() có thể được thưc hiện nhiều lần (ký hiệu bằng hình ellipse).
* Kênh ảo sẽ bị xóa khi Server hoặc Client đóng socket bằng lệnh close().

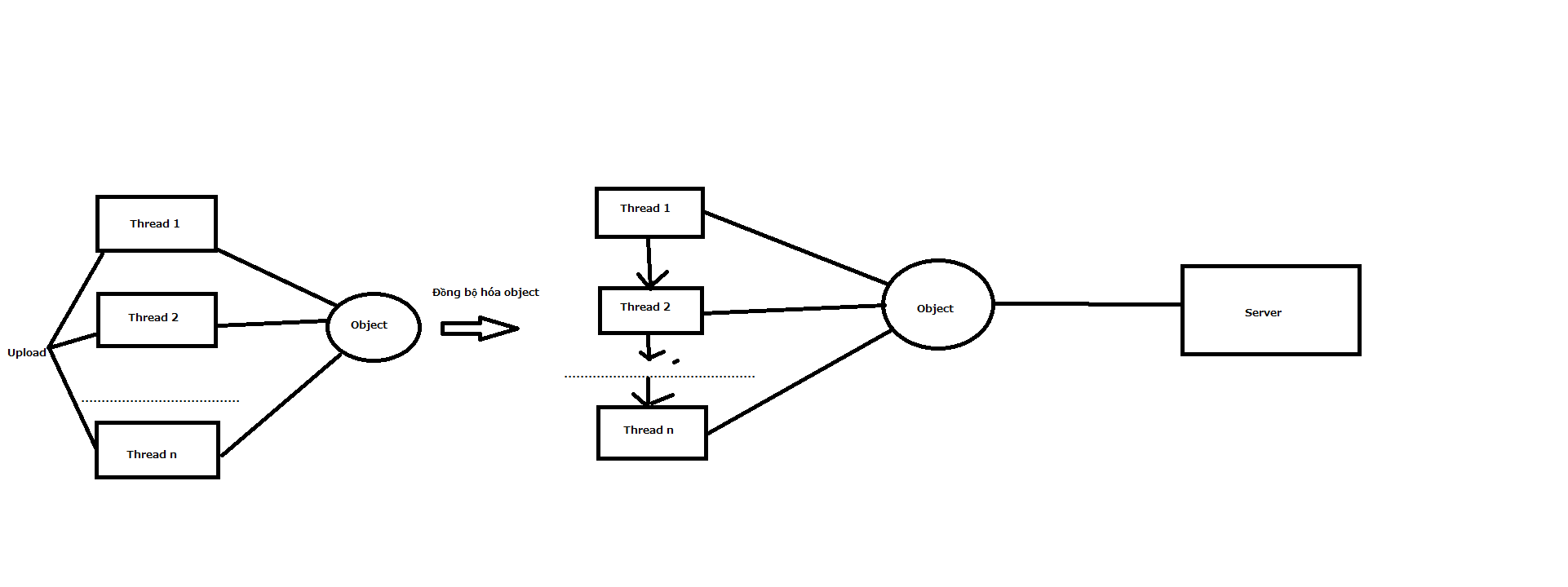
### **Phân tích bài toán**

#### **2.1. Về mặt nghiệp vụ :**

* + Client ( người sử dụng ) có thể upload các tệp tin dữ liệu : hình ảnh, video, mp3,……. lên Server
  + Client ( người sử dụng ) có thể download các tệp tin dữ liệu được lưu ở phía Server về máy cá nhân Client.
  + Trường hợp **upload** lên Server :
* Data tệp tin Client muốn uploadmlene Server thì được phân thành từng mảnh data rồi gửi những mảnh file nhỏ đó lên Server.
* Server nhận được các mảnh đó tiếp tục ghép các mảnh lại và lưu ở tệp dữ liệu đó ở phía Server.
  + Trường hợp **download** về Client :
* Data Server mà Client muốn download về máy được phân mảnh data nhỏ rồi cho phép tải xuống Client
* Client nhận được các mảnh file đó tiếp tục ghép các mảnh lại và lưu ở Client.
  + Client có thể chọn các chức năng :
* **Stop** upload và download khi chưa hoàn thành nhiệm vụ.
* **Pause** upload và download khi chưa hoàn thành nhiệm vụ.
* **Resume** upload và download khi chưa hoàn thành nhiệm vụ.
  + Xử lý tương tranh dữ liệu khi Client upload nhiều data 1 lúc lên Server và dowload nhiều data từ Server về Client.

#### **2.2. Về mặt xử lý :**

* + Cơ chế phân mảnh khi upload, download : Sử dụng **FileInputStream** để phân mảnh tệp tin thành một mảng bytes.
* Lớp **FileInputStream** trong java đọc được các byte từ một input file.
* Nó được sử dụng để đọc dữ liệu theo định dạng byte (các byte stream) như dữ liệu hình ảnh, âm thanh, video vv.
  + Cơ chế ghép mảnh khi upload, download : Sử dụng **FileOutputStream** để ghép mảng bytes thành một file.
* **FileOutputStream** là một output stream được sử dụng để ghi dữ liệu vào một file theo định dạng byte.
* Ghi dữ liệu theo định dạng byte hoặc định dạng ký tự thông qua lớp **FileOutputStream**.
  + Stop upload, download : Sử dụng phương thức **stop();** hủy bỏ một Thread.
  + Pause upload, download : Sử dụng phương thức **suspend();** tạm dừng một Thread.
  + Resume upload, download : Sử dụng phương thức **resume();** tiếp tục một Thread.
  + Xử lý tương tranh dữ liệu khi upload, download : Xử lý bằng cơ chế Đồng bộ hóa đa luồng trong Java ( Đồng bộ hóa Object )



Hình 2 : Cơ chế xử lý tương tranh

* Sử dụng từ khóa **synchronized(object)** trước khi thực hiện các lệnh.
* Ở cách đồng bộ này thì hàm **synchronized** sẽ khóa đối tượng object lưu thông tin cần thiết gửi cho Server ( upload ) hoặc gửi cho Client ( download ).
* Chỉ cho phép tại 1 thời điểm chỉ có 1 luồng được phép truy cập và sử dụng object này.
* Khi thực hiện xong upload, download object thì object đó sẽ được giải phóng để các Thread khác có thể truy cập được.

#### **2.3. Về cấu trúc**

Ứng dụng bao gồm :

* Thư mục Server : Lưu trữ dữ liệu các tệp tin trên Server.
* Thư mục Client : Lưu trữ dữ liệu các tệp tin trên máy người dùng Client.
* Packet MyFTPStorage :
* Controller.java : Xử lý ghép file, chia file và các chức năng ứng dụng
* Data.java : Mô tả đối tượng dữ liệu cần gửi đi.
* Packet View :
* ServerFrame : Giao diện Server.
* ClientFrame : Giao diện Client.

### **Kết quả đạt được và hướng phát triển**

Từ project, chúng em đã có được khả giải quyết yêu cầu bài toán gửi nhận tệp tin và hiểu được cách thức hoạt động của việc gửi nhận tệp tin.

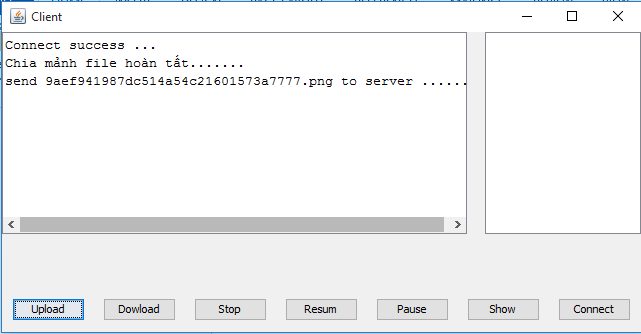
Tuy nhiên hệ thống còn nhiều hạn chế :

1. Chưa upload được file dung lượng quá lớn.
2. Chưa download được file dung lượng quá lớn.

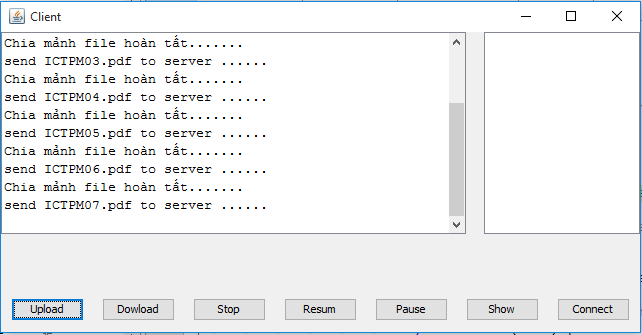
Từ đó, hướng phát triển của hệ thống cần khắc phục các nhược điểm này. Mong rằng nhóm em hoặc nhóm khác kế thừa và phát triển hệ thống này.

# **PHẦN 3 : CÀI ĐẶT VÀ DEMO CHƯƠNG TRÌNH**

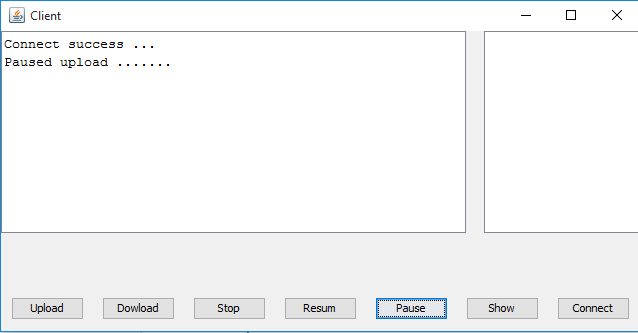
* Chương trình dùng IDE Netbean, java socket
* Một số hình ảnh chương trình :



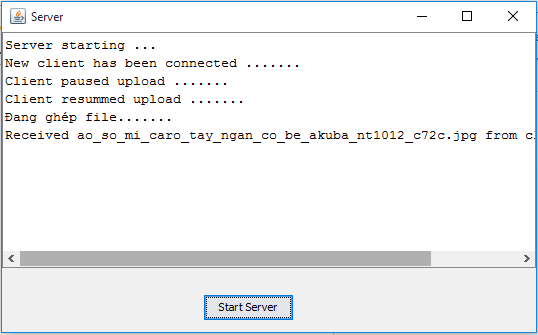
Hình 3 : Client upload lên Server



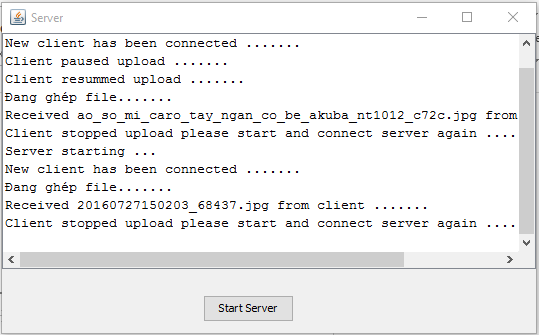
Hình 4 : Upload nhiều tệp tin cùng lúc



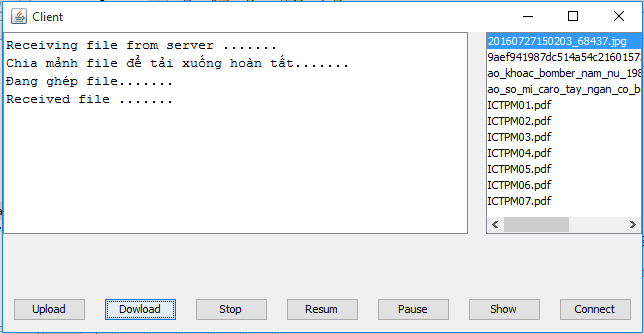
Hình 5 : Pause upload



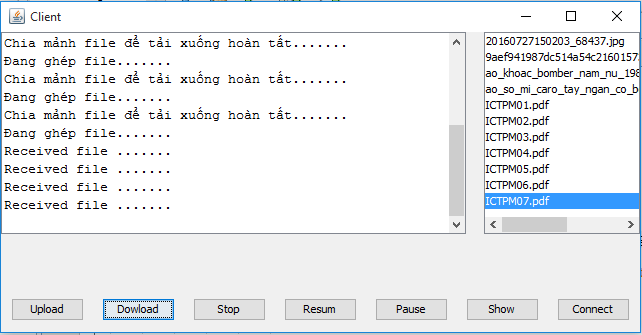
Hình 6 : Resume upload



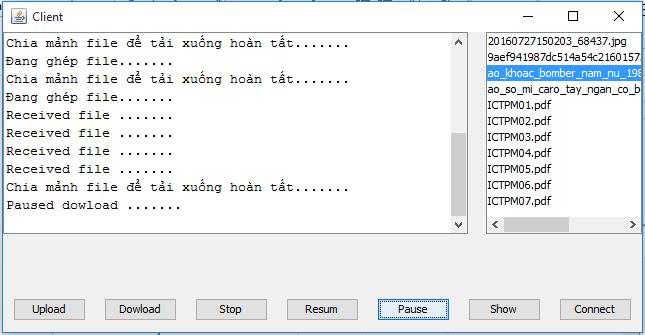
Hình 7 : Stop upload



Hình 8 : Download



Hình 9 : Download nhiều tệp cùng lúc



Hình 10 : Pause download

# **PHẦN 4 : TỔNG KẾT**

### **Danh sách tài liệu tham khảo**

* Slide Tính toán phân tán TS. Nguyễn Hữu Đức.
* Video Youtube : <https://www.youtube.com/watch?v=IMMebNpmSXM>
* Đồng bộ hóa đa luồng trong java : <https://viblo.asia/p/dong-bo-hoa-da-luong-trong-java-ZjlvalBxkqJ>