**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN**

**CHUYÊN ĐỀ HỆ THỐNG PHÂN TÁN**

**Thành phố Hồ Chí Minh**

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN**

**| Đề tài |**

**Cài đặt thuật toán SES**

**| Giáo viên hướng dẫn |**

**Thầy Dũng Trần Trung**

**Thầy Lê Giang Thanh**

**Thành phố Hồ Chí Minh**

**LỜI CẢM ƠN**

Để có thể hoàn thành đồ án này, chúng em xin gửi lời cảm ơn chân thành tới thầy Dũng Trần Trung, thầy Lê Giang Thanh khoa Công Nghệ Thông Tin, trường Đại học Khoa học Tự nhiên vì đã hỗ trợ tụi em trong suốt quãng thời gian làm đồ án, giúp đồ án này được hoàn thiện nhất có thể.

Do thời gian có hạn cũng như các kiến thức của chúng em vẫn có nhiều hạn chế, rất khó để tụi em tránh khỏi những sai sót khi hoàn thiện đồ án cũng như làm báo cáo. Vì vậy rất mong có thể nhận được những lời góp ý từ các thầy cô để chúng em có thể có thêm kiến thức để hoàn thành những đồ án tiếp theo.

Chúng em xin chân thành cảm ơn.

Mục lục

[1. Giới thiệu: 7](#_Toc130149535)

[2. Thành viên và bảng phân công: 7](#_Toc130149536)

[3. Mô tả chi tiết hệ thống: 7](#_Toc130149537)

[4. Chức năng chi tiết của từng Class: 8](#_Toc130149538)

[Sád 8](#_Toc130149539)

[Đa 8](#_Toc130149540)

[ada 8](#_Toc130149541)

**BÁO CÁO**

# Giới thiệu:

* Chương trình chạy trên ngôn ngữ Java. Khi chạy chương trình sẽ tạo ra 7 process và tự động gửi tới 6 process còn lại 1600 messages.
* Mỗi messages có nội dung là “message 1”, “message 2”, …
* Thời gian các message gửi đi sẽ là ngẫu nhiên. Cũng chính vì thế sẽ có hiện tượng buffer và thông tin về timestamp, buffer, delivery hay trạng thái của messages cũng sẽ được in ra terminal và file log.
* Dưới đây là cách cài đặt thuật toán SES (Schiper-Eggli-Sandoz).

# Thành viên và bảng phân công:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Nhiệm vụ** |
| 1 | Nguyễn Gia Minh | 19126054 |  |
| 2 | Lê Trung Hiếu | 19126044 |  |

# Mô tả chi tiết hệ thống:

Hệ thống được tổ chức gồm 4 class chính dưới đây:

* **main**(): Hàm chính của chương trình. Nó bắt đầu bằng việc tạo một object Registry để tạo ra một máy chủ RMI (Remote Method Invocation cho phép các tiến trình chạy trên các máy khác nhau giao tiếp với nhau) và sau đó tạo một số process SchiperEggliSandoz và các class MyProcess. Mỗi process gửi messages đến các process khác bằng MyProcess().
* **MyProcess**: Đây là class dùng Runnable và đại diện cho tiến trình được thực thi. Nó có đầu vào là ( chiperEggliSandoz, danh sách các ID process đích tới, các messages, delayTime). Trong phương thức run(), nó gửi các messages đến các process đích đến.
* **Message**: Lớp này đại diện cho một messages được gửi đi. Nó bao gồm một chuỗi thông điệp, một List<S> và một int[ ] timeStamp.
* **S**: Class này được sử dụng trong việc triển khai đồng bộ hóa tiến trình bằng thuật toán SES. Nó được truyền input gồm (ID, timeStamp). Các function của lớp này cho phép setup ID process và timeStamp và sao chép object S.
* **SBuffer**: định nghĩa các phương thức để dùng cho class S.
* **VectorClock** dùng để đồng bộ hóa thời gian được sử dụng trong hệ thống và theo dõi các sự kiện xảy ra trong các mốc thời gian khác nhau.

# Chức năng chi tiết của từng Class:

## main()

Khi bắt đầu chương trình sẽ khởi chạy hàm main().

Hàm này sẽ bắt đầu tạo ra các process và gửi tin nhắn giữa chúng. Đầu tiên, nó tạo ra một object Registry với port 8080.

Sau đó, nó tạo ra các thứ sau:

* Mảng 2 chiều destIDs để lưu trữ danh sách các Process nơi mà cần gửi messages.
* Mảng 2 chiều Messages để lưu trữ các tin nhắn cần gửi
* Mảng 2 chiều thời gian delays (độ trễ giữa việc gửi các tin nhắn, ở đây là Random).

Sau đó, nó tạo ra một số luồng MyProcess để gửi các tin nhắn theo mảng destIDs được input. Cuối cùng, nó start tất cả các luồng đó.

## class myProcess

Class này thực hiện gửi các tin nhắn giữa các quy trình. Mỗi object MyProcess là 1 process và sẽ gửi tin nhắn đến một hoặc nhiều quy trình khác (ở đây là gửi broadcast cho tất cả process còn lại).

Các tham số đầu vào là (destIDs, messages, delays và process).

Trong hàm run(), vòng lặp đầu tiên duyệt qua các phần tử của mảng destIDs.

Đối với mỗi phần tử, vòng lặp thứ hai hàm process.send() sẽ được sử dụng để gửi thông điệp đến process với ID tương ứng với messages và thời gian delay như đã input vào.

Nếu có lỗi xảy ra, thông báo lỗi sẽ được in ra màn hình vì đã có try, catch error.

## class Message

## d) class S

## e) class SBuffer

## f) class VectorClock