Tên: Phan Ngọc Hạnh Nhi

MSSV: 2131209002

**Practice Lab 4\_2**

For each of the following problems,

1. Identify the correctness constraints of the problem

2. Specify the conditions that each method must wait for

3. Write down the shared state that you will use to check these conditions

## **Assignment 1**

1. **Correctness Constraints (Ràng buộc tính đúng đắn của bài toán)Số lượng tài nguyên hạn chế:**

Semaphore có số lượng permits xác định trước. Một thread chỉ có thể tiếp tục nếu còn permits khả dụng.

* Cơ chế chờ FIFO: Nếu permits == 0, các thread phải chờ theo thứ tự First In, First Out (FIFO).
* Không vượt quá số lượng tài nguyên ban đầu: Khi release() được gọi, số permits không thể vượt quá số ban đầu.
* Tránh deadlock: Mọi thread gọi acquire() đều phải được đánh thức đúng theo thứ tự nếu có release().
* Không có starvation: Không có thread nào bị bỏ qua hoặc bị chặn vô thời hạn.

**2. Conditions that each method must wait for (Điều kiện mà mỗi phương thức phải chờ)**

**Phương thức acquire()**

* Chỉ tiếp tục nếu còn permits (permits > 0).
* Nếu không còn permits, thread phải chờ trong hàng đợi FIFO.
* Sau khi được đánh thức, thread phải giảm permits đi 1.

**Phương thức release()**

* Nếu có thread đang chờ, phải đánh thức thread đầu tiên trong hàng đợi FIFO.
* Nếu không có thread nào chờ, permits được tăng lên nhưng không vượt quá giá trị ban đầu.

**3. Shared State (Trạng thái dùng chung để kiểm tra điều kiện chờ)**

**Các biến trạng thái được sử dụng để kiểm soát semaphore:**

* permits: Số lượng tài nguyên có sẵn. Nếu permits == 0, thread phải chờ.
* maxPermits: Giá trị tối đa của permits, đảm bảo không bị vượt quá khi release().
* queue (FIFO Queue - LinkedList<Condition>): Hàng đợi FIFO để lưu các thread đang chờ, đảm bảo đánh thức theo thứ tự đúng.
* lock (ReentrantLock): Cơ chế khóa để đảm bảo chỉ một thread có thể thay đổi trạng thái của semaphore tại một thời điểm.

## **Assignment 2**

1. **Tạo một class để tạo hợp chất nước (H2O).**

* Chương trình có hai loại thread: **hydrogen (H)** và **oxygen (O)**.
* Khi có **2 nguyên tử hydro và 1 nguyên tử oxy**, chúng sẽ kết hợp lại để tạo thành **H₂O**.
* Nếu chưa đủ số lượng cần thiết, các thread phải **chờ**.

1. **Cơ chế hoạt động của hydrogen()**

* Nếu **số lượng hydrogen >= 1 và oxygen >= 1**, nghĩa là:
  + **1 hydro đã có sẵn + 1 hydro vừa gọi + 1 oxy đã có** → **Đủ điều kiện tạo thành H₂O**.
  + Khi đó, **giảm số H và O đi, sau đó tạo H₂O và đánh thức các thread khác**.
* Nếu chưa đủ điều kiện, **tăng số lượng hydrogen và cho thread H chờ**.

1. **Cơ chế hoạt động của oxygen()**

* Nếu **số lượng hydrogen >= 2**, nghĩa là:
  + **2 hydro đã có sẵn + 1 oxy vừa gọi** → **Đủ điều kiện tạo thành H₂O**.
  + Khi đó, **giảm số H và O đi, sau đó tạo H₂O và đánh thức các thread khác**.
* Nếu chưa đủ điều kiện, **tăng số lượng oxygen và cho thread O chờ**.

1. **Sử dụng số lượng hydrogen và oxygen để kiểm tra điều kiện chờ.**

* Khi một thread H hoặc O gọi phương thức, nó sẽ **kiểm tra số lượng H và O trước khi tiếp tục chạy**.
* Nếu đủ nguyên tử, **tạo H₂O và đánh thức các thread chờ**.
* Nếu chưa đủ, **thread sẽ đợi (await) cho đến khi có đủ số lượng**.