**1.Synchronous vs Asynchronous:**

**a.Synchronous(Đồng bộ):**

* Là khi một tác vụ được thực hiện tuần tự. Quá trình thực hiện sẽ không chuyển sang tác vụ tiếp theo cho đến khi tác vụ hiện tại hoàn thành.
* Mô hình này dễ hiểu và đơn giản, nhưng khi tác vụ nào đó tốn thời gian (như truy vấn dữ liệu từ cơ sở dữ liệu hay đọc file lớn), chương trình sẽ bị "chặn" và không thể thực hiện bất kỳ tác vụ nào khác cho đến khi nó hoàn thành.

**b.Asynchronous (Bất đồng bộ)**:

* Là khi một tác vụ được thực hiện mà không cần đợi tác vụ trước đó hoàn thành. Khi một tác vụ bất đồng bộ được kích hoạt, chương trình có thể thực hiện các tác vụ khác trong khi chờ kết quả từ tác vụ bất đồng bộ.
* Điều này giúp cải thiện hiệu suất, đặc biệt khi bạn phải thực hiện các thao tác tốn thời gian như gọi API hay xử lý file lớn.
* Bất đồng bộ cho phép chương trình tiếp tục chạy mà không bị "chặn"

**2.Các trường hợp sử dụng:**

**a.Synchronous:**

**Trường hợp sử dụng:**

* Khi các tác vụ cần được thực hiện theo thứ tự, tức là kết quả của tác vụ trước phụ thuộc vào tác vụ sau.
* Khi không có các tác vụ tốn thời gian cần xử lý, hoặc môi trường không hỗ trợ tốt bất đồng bộ.

**Ưu điểm:**

* Dễ triển khai và dễ hiểu.
* Không cần phải quản lý trạng thái hoặc xử lý các callback phức tạp

**Nhược điểm:**

* Tốn thời gian chờ khi xử lý các tác vụ như truy vấn csdl,gọi API,…
* Gây ra blocking ảnh hưởng đến hiệu xuất tổng thể khi có các tác vụ lâu hoàn thành.

**b.Asynchronous:**

**Trường hợp sử dụng:**

* Khi bạn phải xử lý các tác vụ lâu hoặc có thể bị chặn như đọc file, gọi API, xử lý dữ liệu từ cơ sở dữ liệu.
* Khi cần thực hiện nhiều tác vụ đồng thời và cải thiện hiệu suất hệ thống.

**Ưu điểm:**

* Cải thiện hiệu suất tổng thể của hệ thống bằng cách không chặn các tác vụ khác trong khi chờ xử lý.
* Phù hợp cho hệ thống yêu cầu mở rộng cao và cần xử lý số lượng lớn tác vụ đồng thời

**Nhược điểm:**

* Phức tạp hơn khi quản lý trạng thái,callback,hoặc điều phối tác vụ
* Cần phải cẩn trọng trong việc xử lý lỗi và quản lý đồng bộ hóa dữ liệu khi nhiều tác vụ thực thi đồng thời.

**3.Từ khóa synchronized trong java?**

**synchronized** là từ khóa trong Java được sử dụng để đảm bảo rằng chỉ có một thread có thể truy cập vào một khối mã hoặc phương thức cụ thể tại một thời điểm, nhằm đảm bảo tính toàn vẹn của dữ liệu trong trường hợp nhiều thread truy cập cùng lúc.

**Cách sử dụng:**

Có thể sử dụng với phương thức hoặc mã khối:

* **Synchronized method:** Khi một thread gọi một phương thức được đánh dấu synchronized,nó sẽ khóa đối tượng mà phương thức này thuộc về hoặc khóa lớp.
* **Synchronized block:**Khối mã được đánh dấu synchronized sẽ khóa một đối tượng cụ thể.Điều này sẽ giúp thu hẹp phạm vi của phần mã được khóa,tối ưu hơn so với việc khóa cả phương thức

**Ưu điểm:**

* Giúp đảm bảo thread safety(an toàn cho thread) bằng cách ngăn chặn nhiều thread cùng truy cập vào tài nguyên chung đồng thời

**Nhược điểm:**

* Hiệu suất có thể bị giảm vì việc sử dụng synchronized có thể dẫn đến các tình huống chờ đợi (blocking) khi nhiều thread cùng cố gắng truy cập vào tài nguyên cùng lúc.
* Có thể gây ra deadlock nếu không sử dụng cẩn thận.