

Swift Multi Thread



Tổng quan

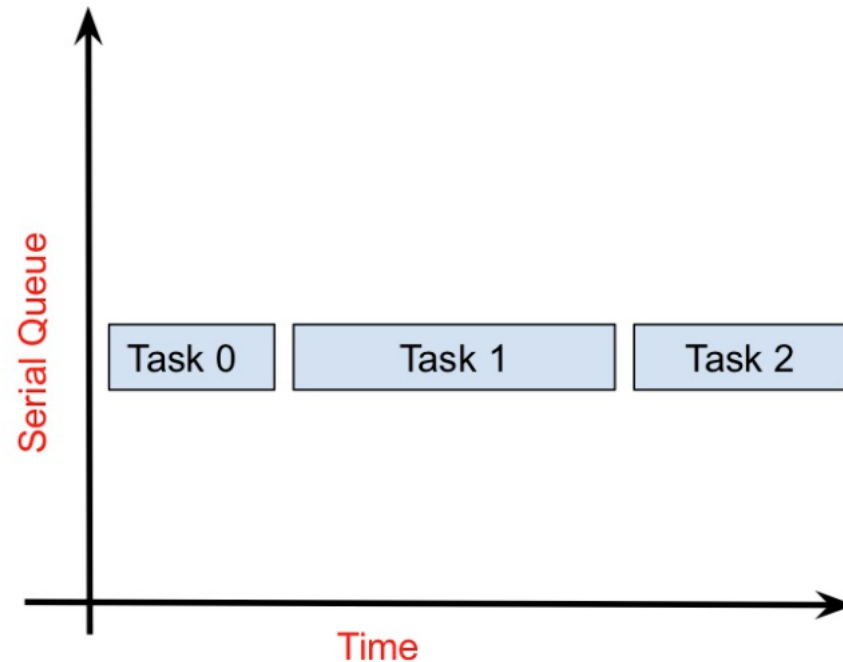
- Process là quá trình hoạt động của một ứng dụng.
- Thread là một luồng bên trong process
- Quản lý đa luồng trong ios: GCD và NSOperation

Grand Central Dispatch (GCD)

- GCD là một API cấp thấp do Apple cung cấp. GCD được xây dựng dựa trên thread nhằm quản lý và chia sẻ thread pool.
 - Nó cải thiện hiệu suất và khả năng đáp ứng của ứng dụng.
 - Ứng dụng sẽ trở nên mượt mà hơn.
 - Thực hiện nhiều nhiệm vụ tại một thời điểm hoặc từng nhiệm vụ theo yêu cầu của bạn, đảm bảo FIFO.
- GCD cung cấp 2 Queue để cung cấp và quản lý các queue (hàng đợi) cho các task :
 - SerialQueue
 - ConcurrentQueue

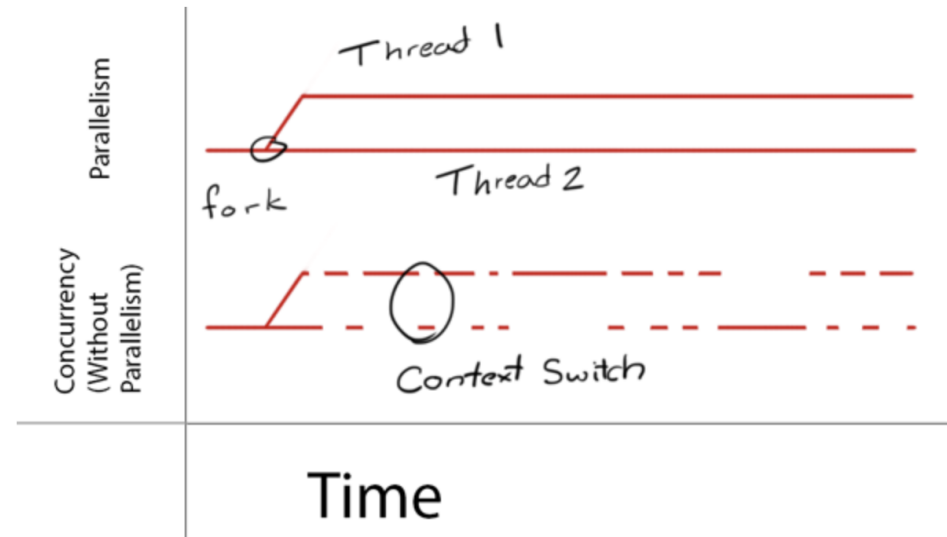
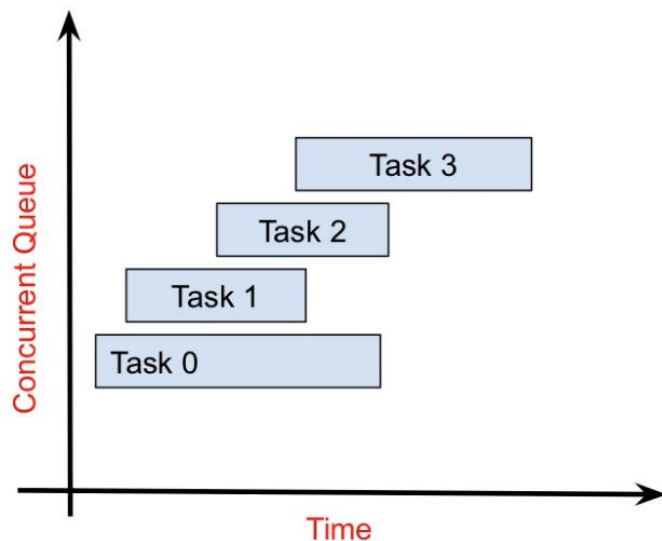
SerialQueue

- Trong Serial Queue, tại một thời điểm chỉ có một task được chạy.
- Sau khi một task kết thúc, task tiếp theo mới được chạy.



ConcurrentQueue

- Trong Concurrent Queue nhiều task có thể chạy tại một thời điểm. Thời gian bắt đầu của task nhiệm vụ sẽ là thứ tự chúng được thêm vào, có nghĩa là task 0 bắt đầu trước sau đó task 1 sẽ bắt đầu sau đó và cứ thế.
- Phân biệt Concurrent và Parallelism



Synchronous vs Asynchronous

- Có 2 cách để gửi một tác vụ cho queue
 - Synchronous : Khi này luồng gọi sẽ phải đợi task kết thúc mới chạy tiếp.
 - Asynchronous : Luồng gọi sẽ chạy tiếp ngay lập tức mà không đợi task kết thúc.

Queue trong IOS

- Main Queue
- Global Queue
- Custom Queue

Main Queue

- Main Queue là SerialQueue
- Tất cả các thao tác update view phải dùng trên MainQueue

```
DispatchQueue.main.async {  
    print("Run on main async")  
}  
print("Run on main after async")
```


Global Queue

- Global Queue là ConcurrentQueue, hay còn được gọi là hàng đợi background. Nó được chia sẻ giữa tất cả ứng dụng.
- Các mức độ ưu tiên (Quality of Service) :
 - background
 - utility
 - default
 - userInitiated
 - UserInteractive
 - Unspecified

Global Queue

```
DispatchQueue.global(qos: .background).async {  
    print("Run on background")  
}
```

```
DispatchQueue.global(qos: .default).async {  
    print("Run on default")  
}
```

Create a Serial Queue

```
let serialQueue = DispatchQueue(label: "mySerialQueue")
serialQueue.async {
    var sum = 0
    for i in 1...1000 {
        sum = sum + i
    }
    print("serialQueue lan 1")
}
serialQueue.async {
    print("serialQueue lan 2")
}
```

Create a Concurrent Queue

```
let concurrentQueue = DispatchQueue(label: "myConcurrentQueue", attributes: .concurrent)

concurrentQueue.async {
    var sum = 0
    for i in 1...1000 {
        sum = sum + i
    }
    print("serialQueue lan 1")
}

concurrentQueue.async {
    print("serialQueue lan 2")
}
```

GCD mở rộng

- Dispatch After
- DispatchSemaphore
- DispatchBarrier
- DispatchGroup

Các vấn đề Multi Thread

- Data race (Thread safe) : Nói về Array và Dictionary trong ios
- Deadlock

NSOperationQueue và NSOperation

- Học buổi sau