

## BÀI 4. TIỀN TRÌNH VÀ XỬ LÝ ĐA TIỀN TRÌNH



### ✓ Mục tiêu:

- Hiểu về tiến trình và cơ chế xử lý đa tiến trình.
- Biết cách sử dụng *Operation Queue* và *Dispatch Queue* để xử lý đa tiến trình.
- Biết cách ứng dụng đa tiến trình vào thực tế

### Bài tập 4.1. Xây dựng ứng dụng sử dụng *Dispatch Queue* để xử lý đa tiến trình

**Đề bài:** Xây dựng mới một ứng dụng có giao diện như sau:



Hình 1.1

Yêu cầu thực hiện:

- Thêm các ảnh đã cho vào Bundle. Các bạn cũng có thể sử dụng các hình khác, tuy nhiên các hình này phải có dung lượng trên 2MB(càng lớn càng tốt) để thấy được khi hiển thị ảnh lên table view sẽ bị chậm và khi scroll sẽ bị giật.

- Sử dụng dispatch queue để chia việc tải ảnh lên table view thành các tiến trình nhỏ nhằm tránh việc tải ảnh ảnh hưởng tới scroll trên table view.

**Mục tiêu:**

- Biết cách sử dụng dispatch queue để xử lý đa tiến trình.

**Gợi ý thực hiện:**

- Dùng dispatch\_async để tạo phiên xử lý riêng.
- Sau khi xử lý xong, dùng thuộc tính dispatch\_get\_main\_queue để trả về luồng xử lý chính.

**Bài tập 4.2. Xây dựng ứng dụng sử dụng Operation Queue để xử lý đa tiến trình**

**Đề bài:** Xây dựng ứng dụng tương tự bài 4.1, tuy nhiên sử dụng Operation Queue để xử lý đa tiến trình thay cho Dispatch Queue



Hình 2.1

**Mục tiêu:**

- Biết cách sử dụng operation queue để xử lý đa tiến trình

**Gợi ý thực hiện:**



- Để khởi tạo phiên xử lý ta sử dụng phương thức `NSOperation` để quản lý các tiến trình.

### **Bài tập 4.3. Áp dụng Operation Queue vào bài tập lớn**

**Đề bài:** Kế thừa mã nguồn của bài tập 2.3, áp dụng Operation Queue để tạo tiến trình xử lý việc tải ảnh cho các màn hình hiển thị danh sách phim đang chiếu, phim sắp chiếu, và chi tiết phim.

**Mục tiêu:**

- Biết cách sử dụng operation queue để xử lý đa tiến trình

**Gợi ý thực hiện:**

- Tạo một đối tượng từ lớp `NSOperationQueue`.
- Sử dụng phương thức **`addOperationWithBlock:`** để thực hiện việc tải hình ảnh trong một thread phụ.