**Очереди**

**Централизованное управление отправкой задач (Grand Central Dispatch)**

**GCD** - технология в iOS и macOS, предназначенная для управления параллелизмом и асинхронностью в многозадачных приложениях. Представляет собой высокоуровневый и удобный интерфейс для создания и управления потоками. Она обеспечивает эффективное распределение задач между доступными ядрами процессора.

**Определение**

*Очереди* представляют собой механизм управления выполнением задач в многозадачной среде. Задачи ставятся в очередь и выполняются в порядке их поступления. Очередь может использовать несколько потоков.

*Очереди в Grand Central Dispatch (GCD)*

Главная очередь (Main Queue): Очередь, связанная с главным потоком. Все обновления пользовательского интерфейса должны быть выполнены на этой очереди.

Глобальные очереди (Global Queues): Предоставляются GCD для выполнения задач в фоновых потоках с различными приоритетами (.background, .utility, .default, .userInitiated, .userInteractive).

Создание собственных очередей: let customQueue = DispatchQueue(label: "com.example.myqueue")

*Очереди в NSOperationQueue*

Главная очередь: Подобно GCD, существует главная очередь для выполнения операций на главном потоке.

**Типы очередей**

*Сериальные(последовательные) очереди (Serial Queues)*

Очередь, выполняющая задачи последовательно в порядке их поступления.

Предотвращает гонки данных при работе с общими ресурсами.

1. let serialQueue = DispatchQueue(label: "com.example.serialQueue")

*Параллельные(согласованные) очереди (Concurrent Queues)*

Очередь, выполняющая задачи одновременно в порядке их поступления.

Увеличивает параллелизм и производительность.

1. let concurrentQueue = DispatchQueue(label: "com.example.concurrentQueue", attributes: .concurrent)

attributes используются для указания дополнительных характеристик параллельных очередей. В частности, атрибуты определяют тип создаваемой очереди и ее поведение.

.concurrent - атрибут указывает, что создаваемая очередь будет параллельной. Это означает, что задачи, добавленные в эту очередь, могут выполняться одновременно в нескольких потоках.

.initiallyInactive - атрибут указывает, что создаваемая очередь будет неактивной, т.е. она не будет выполнять задачи до тех пор, пока не будет активирована вызовом метода activate().

*Очереди синхронизации (Dispatch Barriers)*

Специальный тип очереди GCD, который обеспечивает сериализацию выполнения задач.

Используется для синхронизации доступа к общим ресурсам.

**Группы (Dispatch Groups)**

Позволяют отслеживать завершение группы задач.

Полезны для организации параллельного выполнения задач и оповещения о завершении.

**Ожидание (Dispatch Semaphores)**

Позволяют ограничивать количество потоков, одновременно выполняющихся задачи.

Используются для управления доступом к общим ресурсам.

**Проблемы с очередями**

*Гонки данных (Race Conditions)*

Общий доступ к данным из разных очередей или потоков.

*Взаимные блокировки (Deadlocks)*

Блокировка, когда задачи ожидают друг друга, приводя к замораживанию программы.

*Избыточные ресурсы*

Неправильное использование параллелизма может привести к избыточному потреблению ресурсов.

**ДОПОЛНИТЕЛЬНО**

**Управление потоками**

*NSOperation и NSOperationQueue*

- операции: Абстракция над потоками. let operation = BlockOperation { }

- очередь операций: let operationQueue = OperationQueue()

- добавление операции в очередь: operationQueue.addOperation(operation)

**Синхронизация потоков**

*Операции и блокировки*

- NSLock, NSRecursiveLock, и другие. Используйте для синхронизации доступа к общим данным из разных потоков.