

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ КЕРУВАННЯ ПОТОКАМИ В СЕРЕДОВИЩІ XCODE

Яркун В. І.

Національний університет "Львівська політехніка", студент

I. Постановка проблеми

Одна з тем в програмуванні, яка була актуальною завжди, є багатопотоковість. Правильне застосування інструментів для створення потоків та відповідне розуміння архітектури побудови програми є гарантією успішної її роботи. Проте застосування та керування потоками не є тривіальною задачею, яку легко може вирішити кожен без відповідних навичок та досвіду. Загальні особливості задач передачі в інформаційних системах великих об'ємів даних розглянуто в публікаціях [1, 2]. Багатопотоковість в інформаційних системах є складною задачею, так як потребує постійного використання нових технологій, програмних інструментів та певного досвіду. Основні принципи організації багатопотоковості описано в публікації [3]. Однак при практичній реалізації конкретних рішень є певні труднощі, в першу чергу через брак навичок та досвіду нових користувачів.

II. Мета роботи

Метою дослідження є розробка програмної навчальної бібліотеки керування потоками, для полегшення використання багатопотоковості, при створенні програмного забезпечення, в середовищі Xcode [4].

III. Особливості програмної реалізації бібліотеки

Програмний продукт розроблений у вигляді бібліотеки для середовища Xcode, яка має назву EUThreadManager - Easy to Use Thread Manager (EUTM).

Для створення програмного продукту реалізації бібліотеки EUTM використано об'єктно-орієнтований підхід, а для його розробки – мови Objective-C/Swift, залежно для якого проекту потрібно застосувати дану бібліотеку.

В бібліотеці присутній один клас – EUThreadManager, який спроектований так, що екземпляр даного класу буде створюватись один раз за весь час виконання програми, використовуючи шаблон Singleton [5].

Схема застосування EUTM наведена на рисунку 1. В таблиці 1 наведено особливості інтерфейсу EUTM.

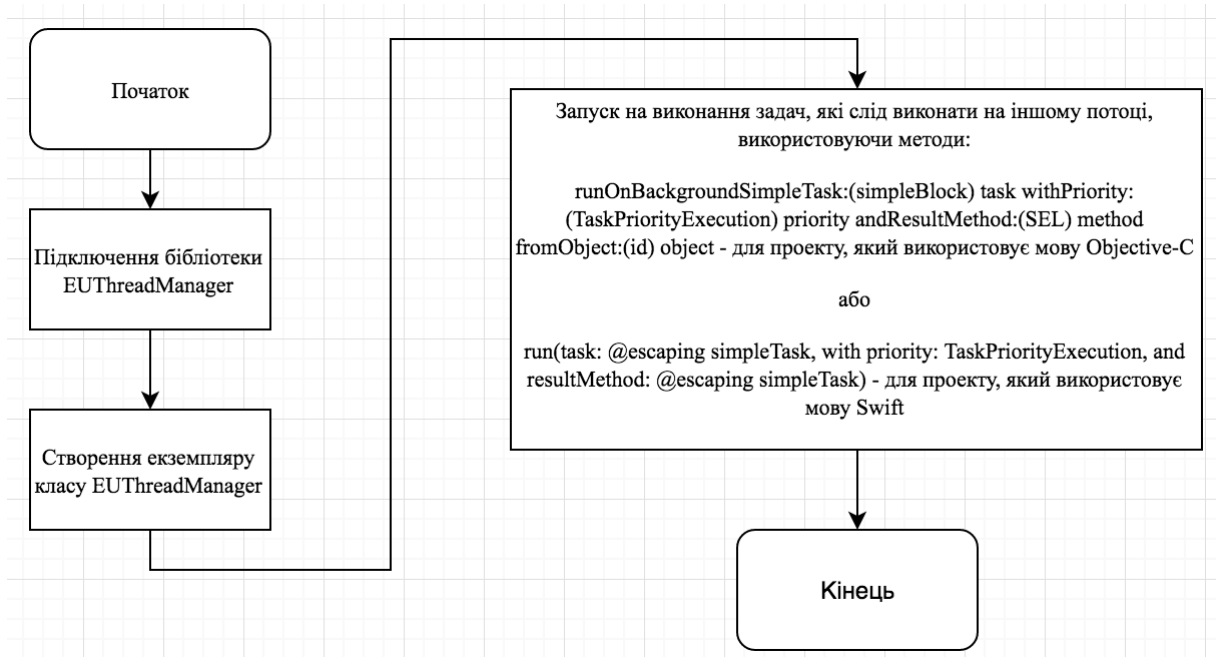


Рисунок 1 – Схема застосування бібліотеки EUTM

Інтерфейс класу EUThreadManager

Objective-C	Swift 3	Опис
<pre>typedef enum : NSInteger { kHighPriorityExecution = 1000, kDefaultPriorityExecution, kLowPriorityExecution, kBackgroundPriorityExecution } TaskPriorityExecution;</pre>	<pre>enum TaskPriorityExecution { case kUserInitiatedPriorityExecution case kDefaultPriorityExecution case kUtilityPriorityExecution case kBackgroundPriorityExecution }</pre>	Варіанти можливих пріоритетів для запуску задач на іншому потоці, відмінному від головного потоку
<pre>+(instancetype) sharedInstance;</pre>	<pre>static let sharedInstance = EUThreadManager()</pre>	Спосіб доступу до екземпляра класу, який буде створено один раз за весь час виконання програми
<pre>-(void) runOnBackgroundSimpleTask:(simpleBlock) task withPriority:(TaskPriorityExecution) priority andResultMethod:(SEL) method fromObject:(id) object</pre>	<pre>run(task: @escaping simpleTask, with priority: TaskPriorityExecution, and resultMethod: @escaping simpleTask)</pre>	Метод для запуску задач на іншому потоці, відмінному від головного потоку

EUTM протестовано для обох мов. При тестуванні було поставлено задачу проходження циклу від 1 до 100000, тестування проводилось на симуляторі Iphone SE, час виконання зайняв 29.591163 с. Отже стільки часу виконувалась задача на іншому потоці, не блокуючи при цьому головний потік. Протягом 29.591163 с програма працювала в мультизадачному режимі, коли користувач міг користуватись іншим функціоналом програми, не боячись її зависання на час виконання вище описаної задачі.

За допомогою бібліотеки EUTM користувач може:

- легко формувати задачі для запуску на іншому потоці [3];
- навчитися самому працювати з інструментами керування потоками на пряму, не використовуючи для цього допоміжні засоби [3].

Висновок

Створено та описано бібліотеку, у форматі навчального програмного продукту, коли користувач здобуває навички керування потоками просто використовуючи EUTM, для запуску задач на інших потоках, відмінних від головного потоку [4]. Наведено короткий опис застосування бібліотеки EUThreadManager, описано архітектуру даної бібліотеки та основних її функцій. Даний програмний продукт є адаптований для двох мов: Objective-C, Swift, які підтримує середовище Xcode, але по аналогії можна адаптувати дану бібліотеку і для інших мов програмування, що в свою чергу дозволить розробникам програмного забезпечення навчитися добре працювати та зрозуміти роботу багатопотоковості у відповідних середовищах програмування.

Список використаних джерел

1. Яркун В. І. Алгоритмічно-програмні засоби синхронізації при обміні даними великих обсягів / В. І. Яркун, Я. С. Парамуд // Вісник Національного університету "Львівська політехніка". Серія: Комп'ютерні системи та мережі : збірник наукових праць – 2016 – №857 – С. 111–118.
2. Яркун В. І. Методика підвищення ефективності інформаційної системи при передачі великих об'ємів даних / В. І. Яркун // Вісник "Тернопільського Національного економічного університету" "Інженерія програмного забезпечення". – 2016 . – С. 173-175.
3. "Pro Multithreading and Memory Management for iOS and OS X with ARC, Grand Central Dispatch, and Blocks" by Kazuki Sakamoto and Tomohiko Furumoto
4. <https://github.com/VitaliyYarkun/EUTM>
5. <https://developer.apple.com/library/content/documentation/General/Conceptual/DevPedia-CocoaCore/Singleton.html>