

УДК 004.41

СРАВНЕНИЕ БЫСТРОДЕЙСТВИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ ПОД ПЛАТФОРМУ IOS

Е.В. КОНЯЕВ

(Представлено: И.С. РУСЕЦКИЙ)

В статье рассматриваются показатели быстродействия языков программирования Swift и Objective-C под платформу iOS.

Ключевые слова: Objective-C, Swift, тестирование, сортировка.

Одной из главных составляющих проектирования под платформу iOS является выбор языка программирования для создания мобильных приложений. Существует два таких языка, со своими характерными особенностями, преимуществами и недостатками. Одним из ключевых параметров является быстродействие – скорость совершения операций с помощью кода написанного на данных языках программирования.

Сравнение быстродействия:

Быстродействие языков тестировалось путем замера скорости сортировок, алгоритмы которых были написаны на всех трех языках. Главный интерес представляют результаты по Swift и Objective-C, классический Си использовался как своего рода критерий. Скорость выполнения написанных на нем программ благодаря низкому, приближающемуся к машинному, уровню абстракции, является очень высокой по сравнению с языками высокого уровня (к категории которых относятся Swift и Objective-C) [1]. Те в меньшей степени учитывают особенности компьютерного железа, удобнее в написании прикладных программ, но по этой же причине медленнее работают.

Результаты испытаний варьируют в зависимости от заданных уровней оптимизации при компиляции исходного кода. Swift значительно уступает Objective-C при отключенной оптимизации, но превзошел его на стандартной оптимизации от 6 до 18 раз, а на высокой — от 7 до 35 раз, в зависимости от типа сортировки [2]. Это очень впечатляющий прирост быстродействия, демонстрирующий явное превосходство Swift по сравнению с Objective-C. На рисунке 1 представлено окно с установкой уровней оптимизации.

▼ Apple LLVM 9.0 - Code Generation	
Setting	FoodTracker
Link-Time Optimization	No ↕
▼ Optimization Level	<Multiple values> ↕
Debug	None [-O0] ↕
Release	Fastest, Smallest [-Os] ↕
Optimization Profile File	/Users/yar/Documents/Xcode/
Relax IEEE Compliance	No ↕
Use Optimization Profile	No ↕

Рисунок 1. – Настройка уровней оптимизации.

Для получения точных данных о быстродействии языков программирования используются различные методы сортировки, такие как:

- Сортировка пузырьком – сравниваются соседние значения и если нужно неравенство не выполняется, то элементы меняются;
- Сортировка выборкой – выполняем поиск наименьшего элемента массива, вставляя его в начало, повторяем эти действия для оставшегося массива;
- Быстрая сортировка – берется случайный элемент массива, удаляется из массива, и массив разбивается на 2 – в первом те элементы что меньше выбранного, во втором остальные. И к каждому из полученных массивов применяются те же действия, пока размер массива не станет нулевым;

— Сортировка слиянием – массив делится пополам, и к каждой половине применяется сортировка слиянием, и после происходит слияние отсортированных половинок за линейное время;

— Стандартная сортировка, которая предполагается стандартной библиотекой.

Быстродействие языков тестировалось путем замера скорости сортировок, алгоритмы которых были написаны на всех трех языках. Главный интерес представляют результаты по Swift и Objective-C, классический Си использовался как своего рода критерий. Скорость выполнения написанных на нем программ благодаря низкому, приближающемуся к машинному, уровню абстракции, является очень высокой по сравнению с языками высокого уровня (к категории которых относятся Swift и Objective-C). Те в меньшей степени учитывают особенности компьютерного железа, удобнее в написании прикладных программ, но по этой же причине медленнее работают.

Таблица 1. – Сравнение быстродействия языков

		Сортировка пузырьком	Сортировка выборкой	Быстрая сортировка	Сортировка слиянием	Стандартная сортировка
Swift	None [-O0]	7,572	2,531	0,077	0,033	0,009
	Fast [-O]	0,326	0,172	0,0222	-	0,0007
	[-Ofast]	0,339	0,180	0,0214	-	0,00007
Obj-C	None [-O0]	2,636	1,189	0,030	0,035	0,0068
	Fastest [-O3]	2,205	0,972	0,029	0,034	0,0069
	[-Ofast]	2,203	0,971	0,030	0,035	0,007
C	None [-O0]	0,457	0,155	0,0044	0,0035	0,0015
	Fastest [O3]	0,163	0,188	0,0038	0,0027	0,0013
	[-Ofast]	0,163	0,186	0,0038	0,0030	0,0013

На Swift при переходе с нулевого уровня оптимизации на оптимизацию внутри каждого файла (средний уровень оптимизации) время работы уменьшается в десятки раз, и в сортировке выборкой он близок к C [3], в то время как на Objective-C время практически не меняется. Здесь явно видно время исполнения, что Swift быстрее чем Objective-C, а C быстрее чем Swift, разве что при использовании стандартной сортировки Swift в 2 раза быстрее.

Заключение. В данной статье было рассмотрено быстроедействие трёх языков программирования, из них два ключевых для создания приложений на платформе iOS. Произведя измерения и анализ полученных результатов, стало понятно, что наиболее производительным и оптимизированным для использования при создании мобильных приложений является язык Swift.

ЛИТЕРАТУРА

1. An analysis of sorts between Swift and Objective-C [Электронный ресурс] // jessesquires.com: [сайт]. URL: <https://www.jessesquires.com/blog/apples-to-apples/>. - Дата доступа: 27.09.2020
2. An analysis of sorts between Swift and Objective-C [Электронный ресурс] // jessesquires.com: [сайт]. URL: <https://www.jessesquires.com/blog/apples-to-apples-part-two/>. - Дата доступа: 27.09.2020.
3. An analysis of sorts between Swift and C [Электронный ресурс] // jessesquires.com: [сайт]. URL: <https://www.jessesquires.com/blog/apples-to-apples-part-three/>. - Дата доступа: 27.09.2020.