

Университет ИТМО
Кафедра ИПМ

Лабораторная работа 1
По предмету:
«Распределенные вычисления»

Выполнил:
Калинин Д. В.
Группа:
Р4115

Санкт - Петербург
2018 г

Задание:

Вариант 2:

Вычисление суммарного количества слов в тексте и нахождение средней длины предложения. Предложения отделяются между собой знаками окончания предложения, а также переносами строк; предложение не может размещаться на нескольких строках одновременно; слова отделяются между собой с помощью знаков препинания и пробела. В результате необходимо вывести ответ в виде одной цифры – кол-ва слов в тексте.

Выполнение:

В качестве исходного текста использовалось произведение «Война и Мир».

Функция чтения текста с диска:

```
func read() -> String? {  
    guard let path = Bundle.main.path(forResource: "war",  
ofType: "txt") else {  
        return nil  
    }  
  
    do {  
        let content = try String(contentsOfFile:path,  
encoding: String.Encoding.utf8)  
        return content  
    } catch _ as NSError {  
        return nil  
    }  
}
```

Затем текст разбивается на параграфы:

```
private func countParagraph(for string: String) -> [String] {  
    return string.characters.split { $0 == "\n" || $0 ==  
"\r\n" }.map(String.init)  
}
```

Функция разделения параграфов на предложения:

```
private func countSentences(for string: String) -> [String] {  
    var r = [Range<String.Index>]()  
    let t = string.linguisticTags(  
        in: string.startIndex..  
string.endIndex,  
scheme: NSLinguisticTagSchemeLexicalClass,  
tokenRanges: &r)
```

```

var result = [String]()
let ixs = t.enumerated().filter {
    $0.1 == "SentenceTerminator"
}.map {r[$0.0].lowerBound}

var prev = string.startIndex
for ix in ixs {
    let r = prev...ix
    result.append(
        string[r].trimmingCharacters(
            in: NSCharacterSet.whitespaces))
    prev = string.index(after: ix)
}
return result
}

```

Функция подсчета количества слов в предложении:

```

private func countWord(in sentence: String) -> Int {
    return
    sentence.components(separatedBy: .whitespacesAndNewlines).filter {
        !$0.isEmpty }.count
}

```

Всем задачам устанавливается высокий приоритет:

```

operationQueue.qualityOfService = .userInitiated

```

Установка числа одновременно выполняемых потоков:

```

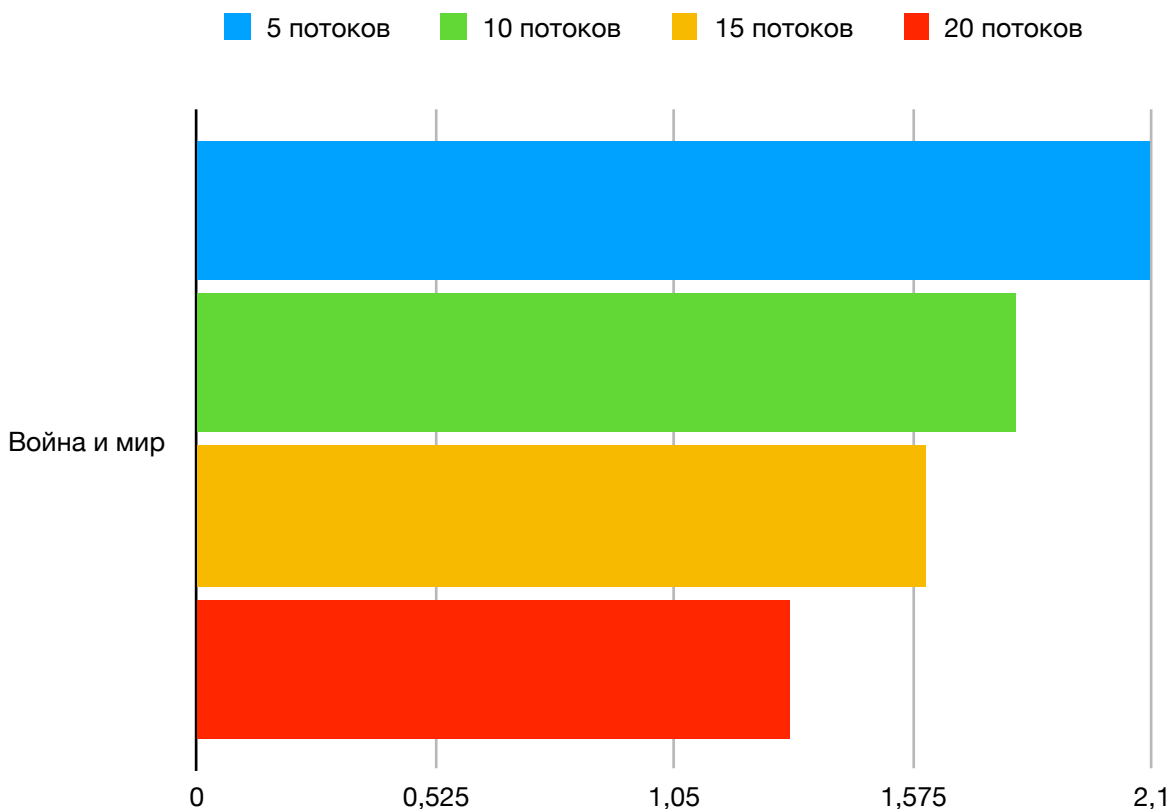
var threadsCount: UInt8 = 3 {
    didSet {
        operationQueue.maxConcurrentOperationCount =
Int(threadsCount)
    }
}

```

Результат:

Подсчет количества слов в предложении оказался «быстрой» задачей, максимальное число ресурсов потребляла задача разделения абзацев на предложения.

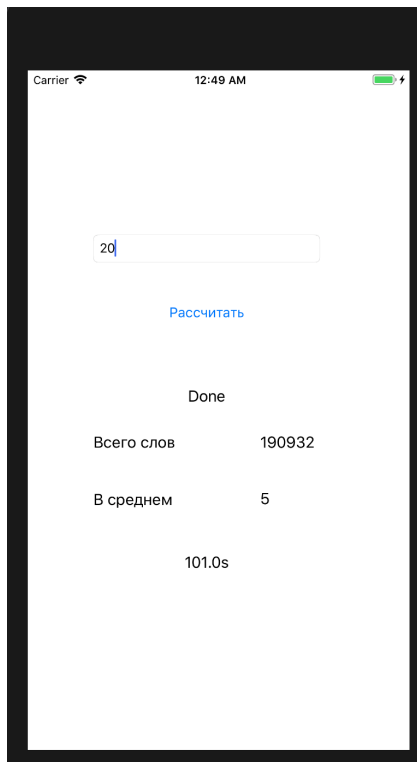
Тем не менее, ниже представлен график зависимости времени выполнения от числа потоков.



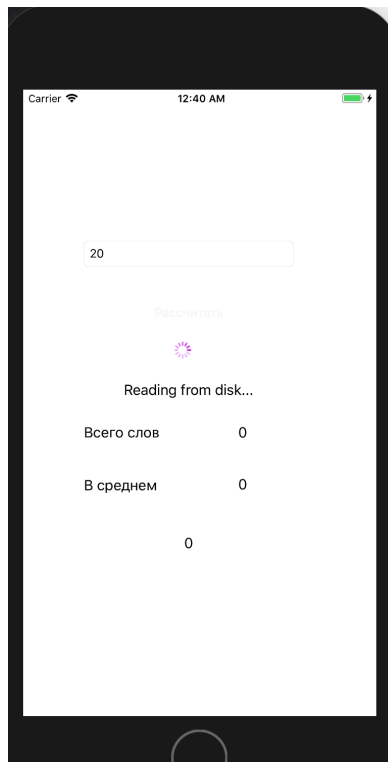
Результат запуска программы:

| | |
|-------------|--------|
| Число строк | 132130 |
| Число слов | 5 |

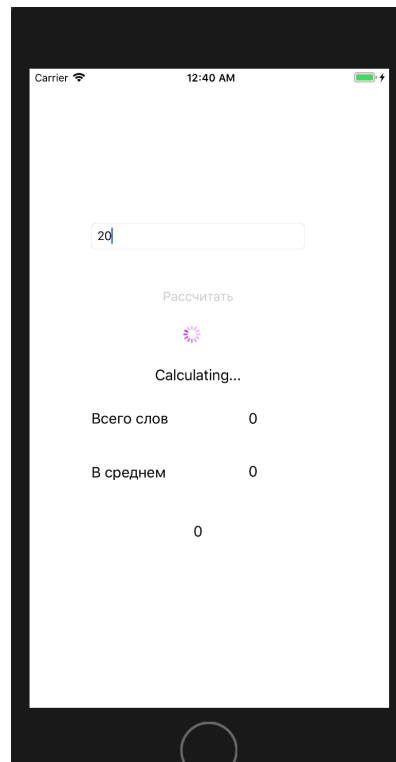
Среднее число слов не совсем соответствует правде так как некоторые символы исходного документа воспринимаются как отдельные предложения и в результате это повлияло на итоговое значение. Это проблема не программы, а исходного документа.



Результат запуска



Процесс чтения с диска



Процесс вычисления