**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи № 4 з дисципліни

«Технології паралельних обчислень»

«Розробка паралельних програм з використанням пулів потоків, екзекьюторів та ForkJoinFramework**»**

**Виконав(ла)**

(шифр, прізвище, ім'я, по батькові)

*ІП-14 Сергієнко Ю. В.*

**Перевірив**

(прізвище, ім'я, по батькові)

*Дифучина О. Ю.*

Київ 2024

Комп’ютерний практикум 4

**Тема:** Розробка паралельних програм з використанням пулів потоків, екзекьюторів та ForkJoinFramework.

**Виконання:**

1. Побудуйте алгоритм статистичного аналізу тексту та визначте характеристики випадкової величини «довжина слова в символах» з використанням ForkJoinFramework. **20 балів**. Дослідіть побудований алгоритм аналізу текстових документів на ефективність експериментально. **10 балів**.

Для виконання даного завдання було використано приклад реалізації пошуку слова у структурі файлів від Oracle. У ньому реалізований однопотоковий пошук (перебиранням файлів) та багатопотоковий пошук з використанням ForkJoinFramework. Основним готовим функціоналом є конвертація тексту усіх файлів у слова з видаленням пунктуаційних символів.

Для знаходження середньої довжини слова було додано клас WordLength, у якому описані avg та count параметри. З кожною ітерацією слів будемо розраховувати avg параметр. Для проведення коректного експерименту було створено структуру файлів із книг у текстовому форматі (рисунок 1).

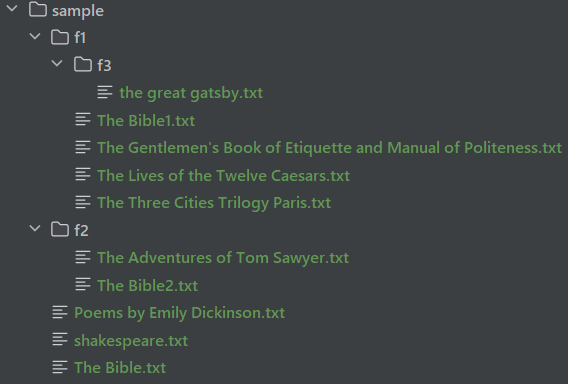


Рисунок 1 – Структура файлів

Результати замірів часу роботи алгоритмів (1 замір – по 10 ітерацій) можна бачити у таблиці 1.

Таблиця 1 – Результати експерименту

|  |  |  |  |
| --- | --- | --- | --- |
| № | Single, ms | Parallel, ms | Acc, t |
| 1 | 1188 | 459 | 2.42 |
| 2 | 1160 | 350 | 3.31 |
| 3 | 1137 | 271 | 4.2 |

1. Реалізуйте один з алгоритмів комп’ютерного практикуму 2 або 3 з використанням ForkJoinFramework та визначте прискорення, яке отримане за рахунок використання ForkJoinFramework. **20 балів**.

Було створено реалізацію алгоритму множення матриць Фокса за допомогою ForkJoin, адже тут можливо розбити на підзадачі основну дію – розбиття матриці на блоки. У рекурсивній задачі MultiplyMatrixTask будуть запускатись q \* q підзадач (де q – к-сть блоків діагоналі), поки к-сть елементів блоку m більша за поріг або поточний блок ще можливо поділити. Коли алгоритм доходить порогу, відбувається множення поточних блоків та повернення даного результату.

Таблиця 2 – Результати роботи алгоритму Фокса

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Size | Threads | Fox, ms | FJ Fox | | Acc, t |
| Threshold | Time, ms |
| 1000 | 4 | 324 | 125 | 294 | 1.1 |
| 16 | 224 | 250 | 1550 | 0.15 |
| 2048 | 4 | 4674 | 128 | 2059 | 2.27 |
| 32 | 2084 | 2.24 |
| 16 | 2556 | 8 | 3195 | 0.8 |
| 32 | 1716 | 1.49 |
| 3000 | 4 | 21256 | 375 | 13268 | 1.6 |
| 16 | 12814 | 750 | 198448 | 0.064 |

У результаті спостережень виявлено, що алгоритм множення з використанням ForkJoinFramework справляється із задачею краще при більшій кількості поділів, але не при найменшій. При мінімальній кількості поділів (при вхідних матрицях, розмір яких неподільний на кількість потоків) більш оптимальним є звичайний алгоритм.

1. Розробіть та реалізуйте алгоритм пошуку спільних слів в текстових документах з використанням ForkJoinFramework. **20** **балів**.

Для виконання даної задачі було взято рішення Oracle для пошуку слів. Щоб знайти спільні слова вхідних документів, необхідно пройти кожний з них, записати усі слова та кількість їх повторів (найкращий варіант – структура Map). Далі, після ітерації, з’єднаємо поточну структуру та знайдену. В результаті отримаємо список слів з к-стями їх повторів (рисунок 2).

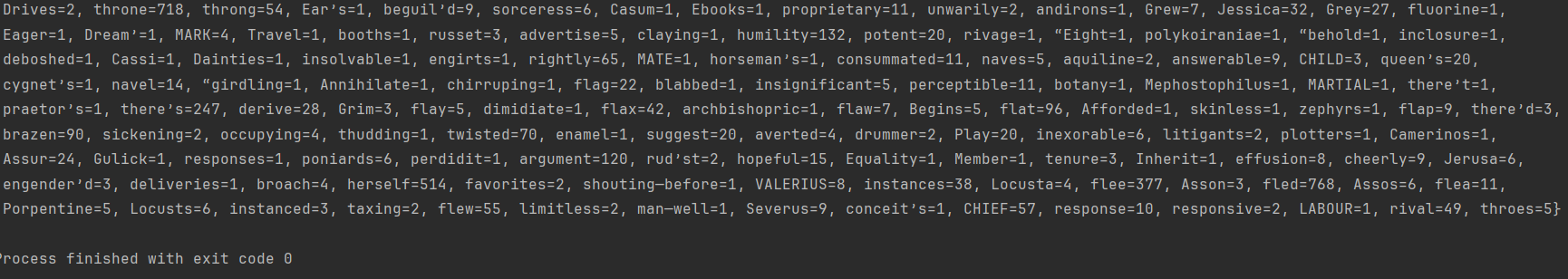


Рисунок 2 – Результат пошуку спільних слів

1. Розробіть та реалізуйте алгоритм пошуку текстових документів, які відповідають заданим ключовим словам (належать до області «Інформаційні технології»), з використанням ForkJoinFramework. **30 балів**.

Щоб виконати цю задачу необхідно модифікувати початковий код, а саме додати перевірку під час ітерації слів документу, чи воно співпадає зі словами області ІТ. Якщо так, будемо додавати назву поточного документу в результуючий список (рисунок 3).

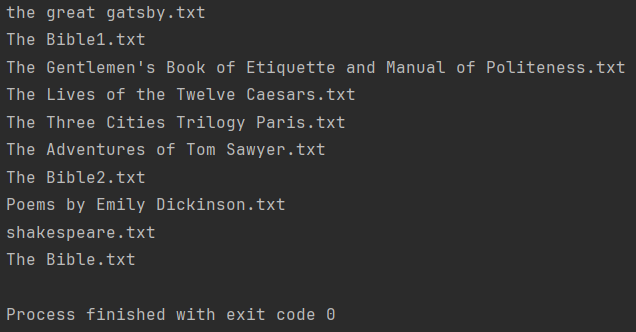


Рисунок 3 – Результуючі документи

**Висновок**

Під час виконання даного комп’ютерного практикуму я закріпив знання та навички щодо розробки паралельних алгоритмів з використанням ForkJoinFramework. Даний фреймворк дає змогу оптимізувати будь-який процес чи алгоритм, який можна перебудувати рекурсивно. На прикладі даних задач бачимо, що шукати по файлах можна значно швидше за допомогою пулу потоків та рекурсії. Алгоритм Фокса з використанням ForkJoin є більш оптимальним на об’ємах даних, розміри яких є подільними, тобто є змога зробити більше розгалудження задач для зменшення навантаження системи.

Код програми доступний на [Github](https://github.com/sergienkoyura/kpi_tpo).