**О Т Ч Е Т**

о выполнении лабораторной работы № 3

по дисциплине «Распределенная обработка информации»

Вариант 14

Выполнил:

Чернов А.А. \_\_\_\_\_\_\_\_\_\_

(ПОДПИСЬ)

Проверил:

Мальцев Р.В. \_\_\_\_\_\_\_\_\_\_

(ПОДПИСЬ)

МОСКВА 2019

Оглавление

[1 Задание 3](#_Toc22561679)

[2 Теоретическая справка 4](#_Toc22561680)

[3 Спецификации разработанных программ 6](#_Toc22561681)

[3.1 Программное средство «Генератор» 6](#_Toc22561682)

[3.2 Программное средство «Однопоточное приложение» 6](#_Toc22561683)

[4 Алгоритм решения задачи 8](#_Toc22561684)

[5 Тестирование программных средств 11](#_Toc22561685)

[5.1 Проверка корректности реализованных приложений 11](#_Toc22561686)

[5.2 Проведение экспериментов 14](#_Toc22561687)

[5.3 Результаты экспериментов 15](#_Toc22561688)

[5.4 Анализ полученных результатов 17](#_Toc22561689)

# Задание

Для заданного каталога построить индекс файлов, представляющий собой отсортированный массив пар (<символ>; <полное имя файла>). Где <символ> – любой из символов встречающийся в сокращенном имени файла (т.е. имени без пути к нему). Подумать над вариантом алгоритма формирующим пары <подстрока>; <полное имя файла>), где максимальная длина подстроки задается при запуске алгоритма построения. Также подумать над вариантом формирования пар со всеми встречающимися в именах файлов вариантами подстрок. Построить индекс файлов виде древовидной структуры.

# Теоретическая справка

Поисковый индекс — структура данных, которая содержит информацию о документах и используется в поисковых системах. Индекси́рование, совершаемое поисковой машиной — процесс сбора, сортировки и хранения данных с целью обеспечить быстрый и точный поиск информации. Создание индекса включает междисциплинарные понятия из лингвистики, когнитивной психологии, математики, информатики и физики. Веб-индексированием называют процесс индексирования в контексте поисковых машин, разработанных, чтобы искать веб-страницы в Интернете.

Популярные поисковые машины сосредотачиваются на полнотекстовой индексации документов, написанных на естественных языках. Мультимедийные документы, такие как видео и аудио и графика, также могут участвовать в поиске.

Метапоисковые машины используют индексы других поисковых сервисов и не хранят локальный индекс, в то время как поисковые машины, основанные на кешированных страницах, долго хранят как индекс, так и текстовые корпусы. В отличие от полнотекстовых индексов, частично-текстовые сервисы ограничивают глубину индексации, чтобы уменьшить размер индекса. Большие сервисы, как правило, выполняют индексацию в заданном временном интервале из-за необходимого времени и затрат на обработку, в то время как поисковые машины, основанные на агентах, строят индекс в масштабе реального времени.

# Спецификации разработанных программ

## Программное средство «Многопоточное приложение с использованием библиотеки Qt»

Пользовательский интерфейс созданного программного средства реализован следующим образом:

mapreduce --input inputFilesName --output outputFileName.csv

где mapreduce – имя исполняемого файла разработанной программы; --input – ключ, определяющий следующий аргумент командной строки как общую часть для имени группы входных файлов, обрабатываемых данных; inputFilesName – общая часть для имени группы входных файлов; --output – ключ, определяющий следующий аргумент командной строки как имя выходного файла с результатами измерений в формате CSV; outputFileName.csv – имя выходного файла. Кроме перечисленных допускается наличие других аргументов командной строки, они должны следовать после обязательных аргументов, перечисленных выше.

# Алгоритм решения задачи

Программа mapreduce требует дополнительные параметры – конфигурационный файл и длина подстрок, на которые необходимо разбить имена входных файлов. Ее результатом является набор выходных файлов с результатами разбиения и файл, содержащий время обработки каждого из входных файлов.

В функции main обрабатываются параметры программы. Затем считывается сформированная генератором «порция» данных. Для каждой «порции» данных выполняются вызовы функции mappedReduced из пространства имен QtConcurrent.

qm\_sub res\_sub = QtConcurrent::mappedReduced(full\_files, build\_subline\_index, merge\_map\_sub);

Для этого требуется подключить основные библиотеки и заголовки библиотеки Qt (QMap, qtconcurrentmap.h, qapplication.h) и задать тип результирующего значения к которому следует преобразовать QFuture - результат работы mappedReduced.

typedef QMultiMap<string, string> qm\_sub;

Функция build\_subline\_index (map) - работа, которую нужно произвести с данными. merge\_map\_sub (reduce) - функция, которая объединяет результат обработки данных и сводит их к необходимому типу данных.

Каждая порция данных обрабатывается параллельно и независимо функциями build\_subline\_index и build\_symbol\_index. По окончанию обработки словари, полученные в результате работы функций, сливаются в один общий (для строкового и символьного индексов - соответственно). Для того, чтобы не вызывать конструктор копирования контейнеров при вызове каждой функции, организована передача по-ссылке.

# Тестирование программных средств

## Проверка корректности реализованных приложений

Были сравнены выходные значения обработки одних и тех же данных при помощи monothread, multithread, mapreduce. Результаты соответствуют ожиданиям.

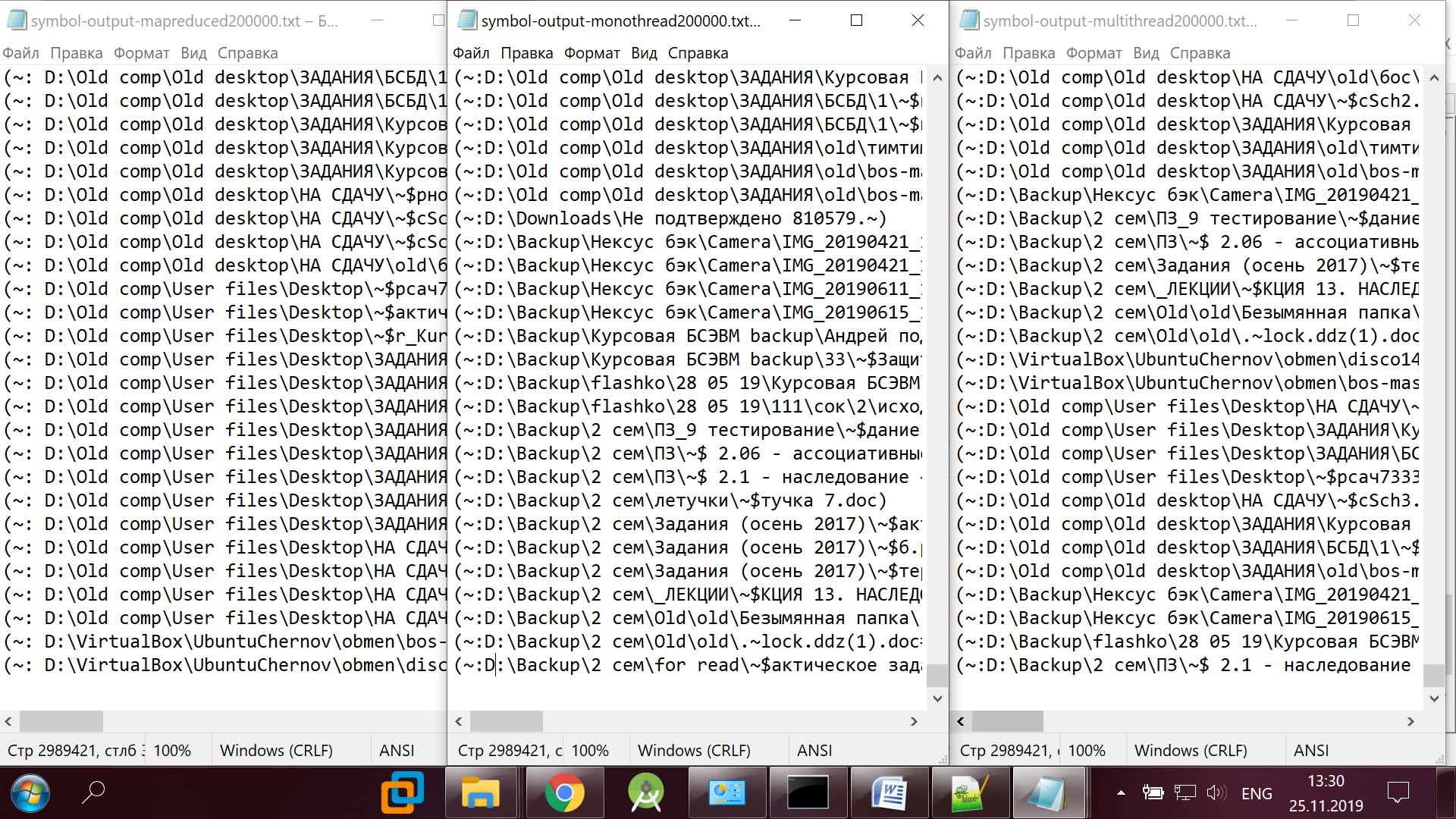


Рисунок 1 – входные monothread, multithread, mapreduce символьный индекс

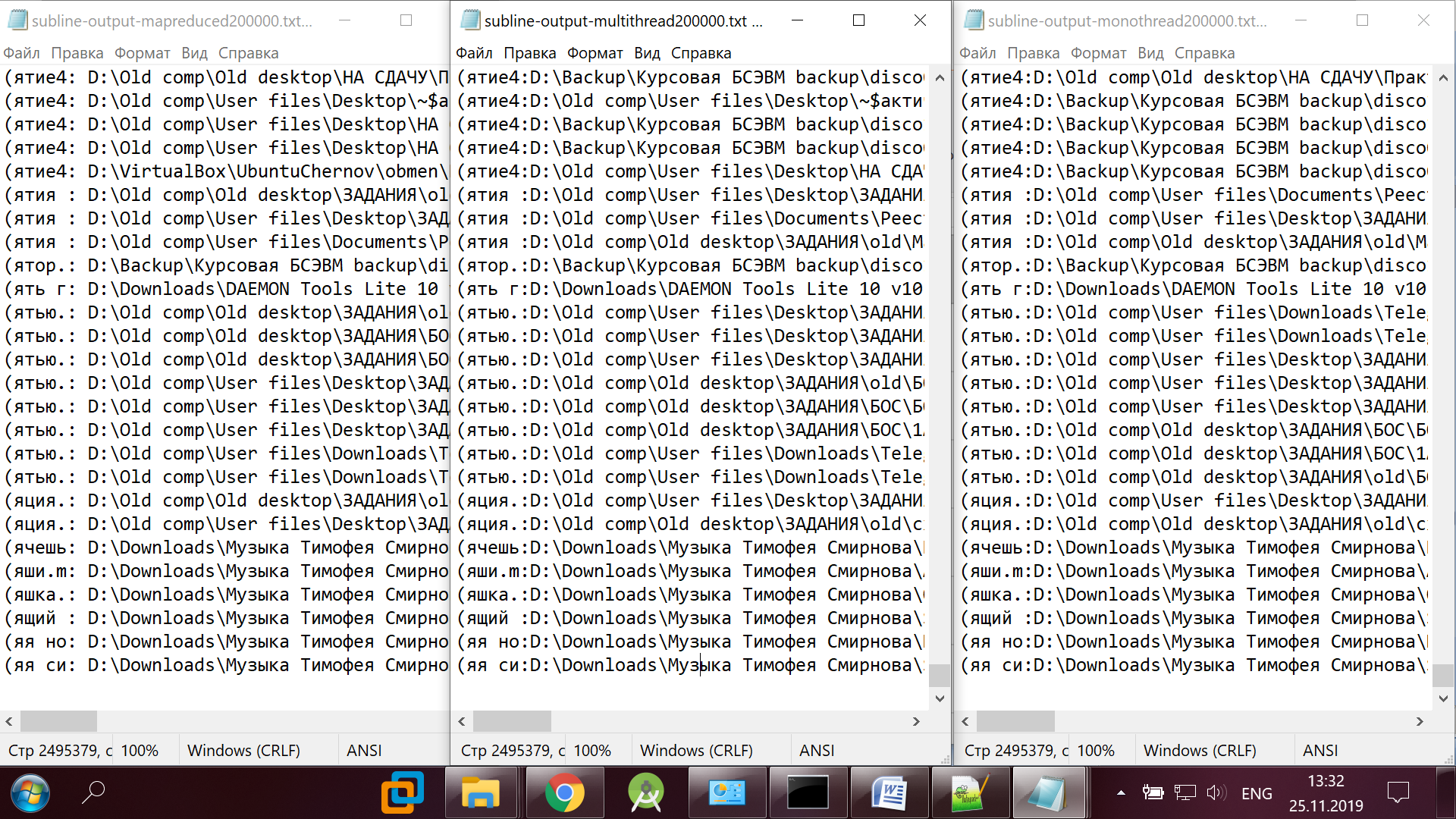


Рисунок 2 – входные monothread, multithread, mapreduce строковый индекс

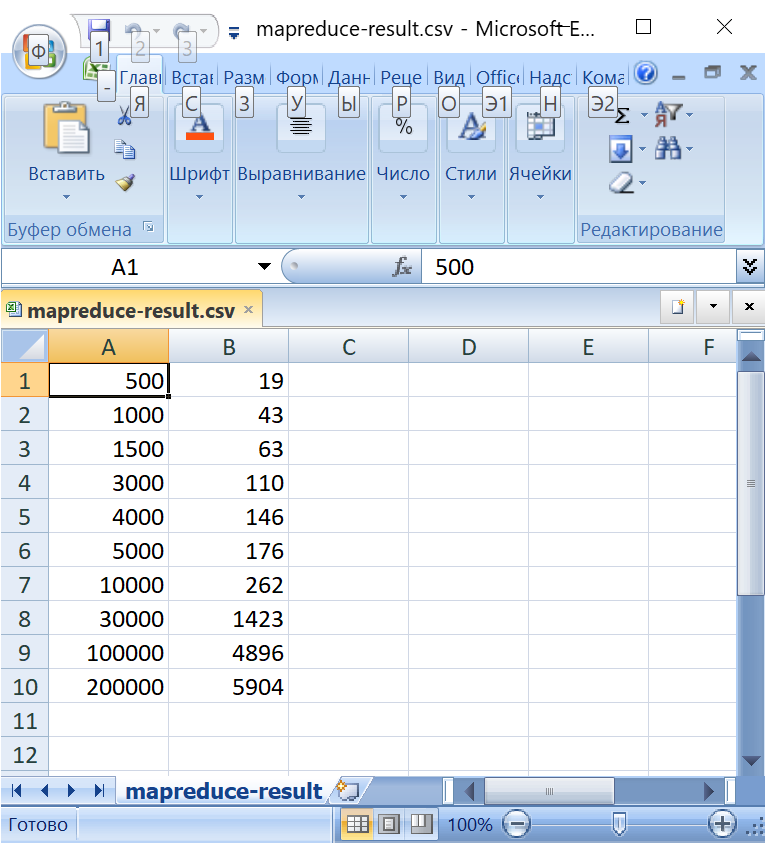


Рисунок 3 – время обработки данных

## Проведение экспериментов

Был проведен эксперимент для большого количества данных. Для программы generator был указан путь «D:/» и сгенерирован большой объем входных данных.

## Результаты экспериментов

Результат выполнения monothread и обработки данных был представлен в таблице 1.

|  |  |
| --- | --- |
| Количество строк | Время (мс) |
| 500 | 19 |
| 1000 | 43 |
| 1500 | 63 |
| 3000 | 110 |
| 4000 | 146 |
| 5000 | 176 |
| 10000 | 262 |
| 30000 | 1423 |
| 100000 | 4896 |
| 200000 | 5904 |

Таблица 1 – результат эксперимента

## Анализ полученных результатов

В ходе лабораторной работы были реализованы генерация и обработка больших объемов данных. На скорость обработки влияет загруженность компьютера другими процессами и общее количество программ, запущенных в данный момент. Для ускорения обработки была использована библиотека Qt. Были получены результаты превосходящие многопоточный вариант реализации задачи по времени. Ускорения обработки удалось добиться за счет быстрого ссылочного перебора Map - контейнеров, реализованной в QMap. Для каждой записи не вызывается конструктор копирования - что существенно ускоряет программу. Также QtConcurrent исходя из возможностей ПК сама эффективно распределяет работу по ядрам. Это объясняет полученные результаты.

На графике наблюдается близкая к логарифмической зависимость. При больших объемах данных заметна эффективность многопоточного решения. На маленьких объемах расхождения несущественны из-за необходимости слияния результатов обработки. Временные результаты могут отличаться, в зависимости от длины подстрок разбиения и заданной директории для построения индекса.

Рисунок 4 – Оценка сложности