**О Т Ч Е Т**

о выполнении лабораторной работы № 1

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 «Генератор»

Данное программное средство представляет собой консольное приложение, предназначенное для формирования входных файлов для однопоточных и многопоточных приложений.

Пользовательский интерфейс программного средства реализован следующим образом:

generator --config parameters.cfg --output outputFilesName

где generator – имя исполняемого файла программы-генератора, изменять его (задавать по собственному усмотрению) недопустимо, ––config – ключ, определяющий следующий аргумент командной строки как файл параметров генерируемых данных, содержащий, в том числе, перечень объемов генерируемых данных; parameters.cfg – имя файла параметров генерируемых данных; ––output – ключ, определяющий следующий аргумент командной строки как общую часть имени для группы выходных файлов генерируемых данных разного объема; outputFilesName – имя общей части имени для группы выходных файлов генерируемых данных.

## Программное средство «Однопоточное приложение»

Пользовательский интерфейс созданного программного средства реализован следующим образом:

monothread --input inputFilesName --output outputFileName.csv

где monothread – 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

Прогррамное средство генератор помимо обязательных параметров требует передать путь начальной директории для построения индекса файлов в ней. Для решения задачи была использована библиотека boost - filesystem. На ее основе был разработан класс IndexBuilder.

class IndexBuilder

{

public:

IndexBuilder(boost::filesystem::path p);

~IndexBuilder();

std::vector<fs::path> get\_subdirs();

private:

boost::filesystem::path full\_path;

std::vector<fs::path> subdirs;

std::vector<string> files;

std::vector<fs::path> build\_subdirs();

std::vector<string> build\_files();

};

В начале работы программы рекурсивно методом build\_subdirs заполняется вектор subdirs. Он содержит список вложенных директорий. Также в вектор files записываются полные пути файлов, содержащихся в директориях. На основе конфигурационного файла вычисляется, сколько путей необходимо сформировать для данной директории.

Программа monothread также 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.

# Тестирование программных средств

## Проверка корректности реализованных приложений

Была проверена корректность генерации данных программой generator.

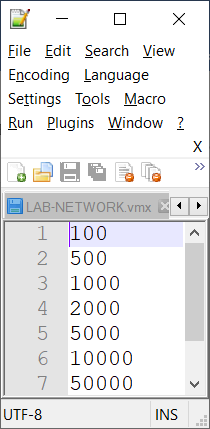


Рисунок 1 – входные данные для генератора

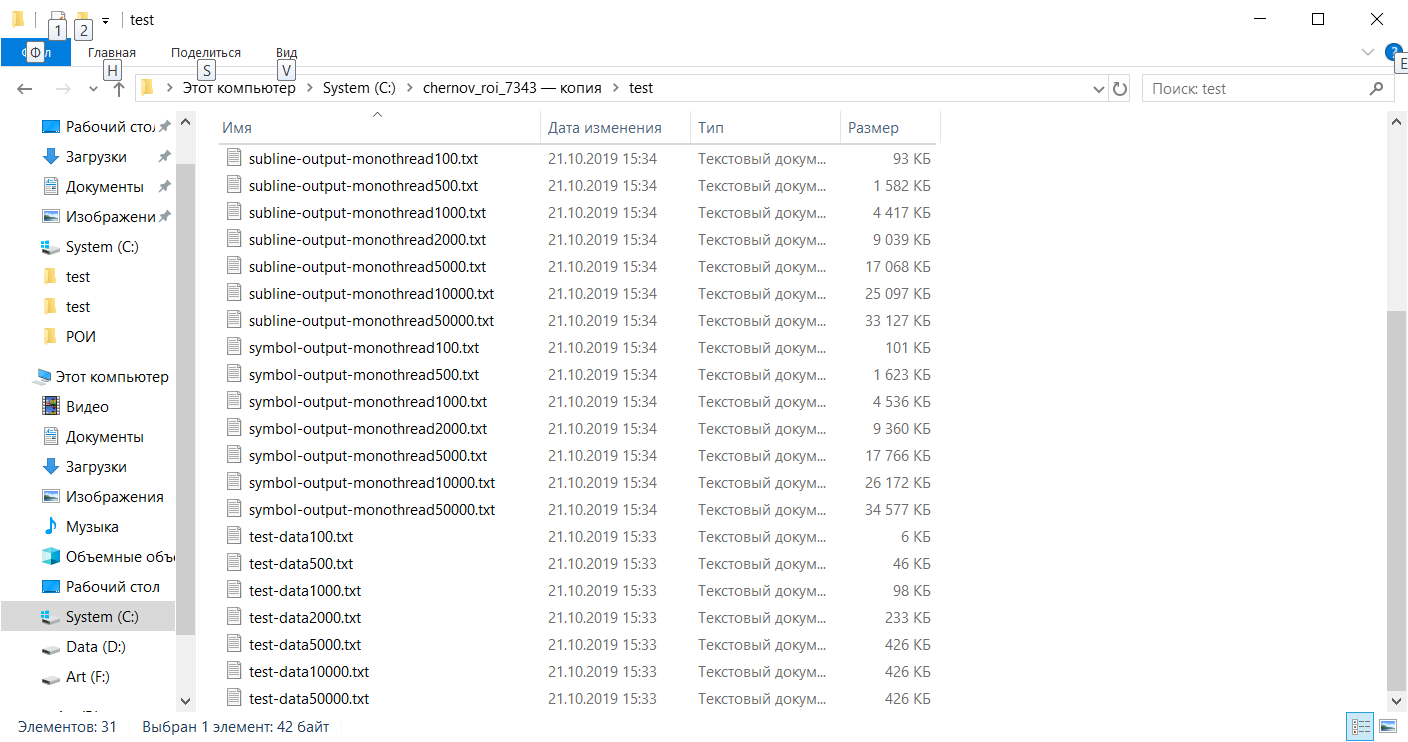


Рисунок 2 – выходные данные генератора

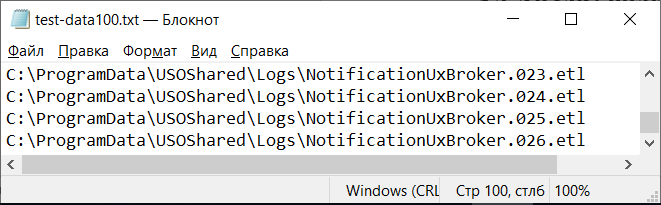


Рисунок 3 – результат генерации 100 тестовых строк

Была проверена корректность обработки данных программой monothread.

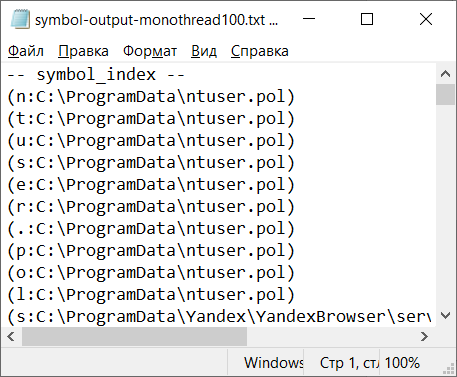


Рисунок 4 – результат построения символьного индекса

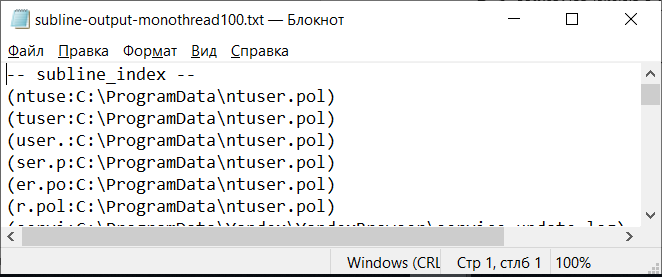


Рисунок 5 – результат построения строкового индекса (длина 5)

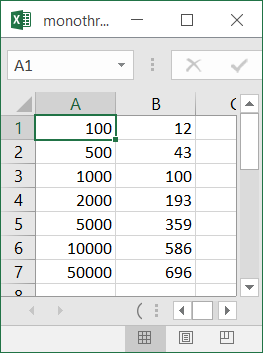


Рисунок 6 – время обработки данных

## Проведение экспериментов

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

## Результаты экспериментов

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

|  |  |
| --- | --- |
| Количество строк | Время (мс) |
| 100 | 2 |
| 500 | 9 |
| 1000 | 19 |
| 2000 | 39 |
| 5000 | 91 |
| 7500 | 145 |
| 10000 | 366 |
| 25000 | 1703 |
| 50000 | 2678 |
| 75000 | 3182 |
| 100000 | 3917 |
| 150000 | 4516 |
| 200000 | 5732 |
| 250000 | 6771 |
| 300000 | 7883 |
| 350000 | 9950 |
| 400000 | 10931 |
| 450000 | 12034 |
| 500000 | 14520 |

Таблица 1 – результат эксперимента

## Анализ полученных результатов

В хоте 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 возможно параллельное выполнение обработки с последующей оптимизацией.

На графике наблюдается близкая к линейной зависимость. Так как в цикле построения индекса перебираются все элементы контейнера. Вложенный цикл перебора букв (подстрок) слова не вносит больших задержек ввиду небольшой длины слова. Также временные результаты могут отличаться, в зависимости от длины подстрок разбиения и заданной директории для построения индекса.

Рисунок 7 – оценка сложности