**DOKUZ EYLUL UNIVERSITY**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CME2101 – Project Based Learning-III**

**TEXT BASED SEARCH ENGINE**

**by**

**Gül Eda Aydemir**

**Seher Balci**

**Ömer Beyazıt**

**November, 2016**

**İZMİR**

**Page**

**CONTENTS**

**CHAPTER ONE**

**INTRODUCTION 1**

**CHAPTER TWO**

**PROJECT DESCRIPTION 2**

**CHAPTER THREE**

**PROGRESS SUMMARY 3**

3.1. Project-Tasks Approach 3

3.2. Time-Periods Approach 3

**CHAPTER FOUR**

**PROBLEMS ENCOUNTERED &CHANGES IN REQUIREMENTS 4**

**CHAPTER FIVE**

**CLASSES…………………………………………………………………………….5**

**CHAPTER SIX**

**COMPARISION HASH METHODS………………………………………………6**

**CHAPTER SEVEN**

**SCREENSHOTS ……………………………………………………………………7**

**CHAPTER EIGHT**

**CONCLUSION & FUTURE WORK………………………………………………8**

**REFERENCES 8**

# CHAPTER ONE

INTRODUCTION

The purpose of the report was to explain the project called ‘Text Based Search Engine’ devoloped for Project Based Learning -III class. The project started in 29th September,2016 . The deadline of the Text Based Search Engine project was 8th November,2016. The Text Based Search Engine project planned as a seven week project.

The aim of the Text Based Search Engine project was to operate on text files that are kept in a single directory. Therefore, user of the project can find the text file by entering words that what the user is looking for.

# CHAPTER TWO

Project Description

The Text Based Search Engine project was developed for a User who wants to find the text file by entering the words which similiar to what is the User looking for.

First of all, a text based files given for this project for testing our Text Based Search Engine Project. After decoding them into UTF8 and removing all StopWords, all the words from all the documents insert into Hash Table. What happened here is called File Operations.

Secondly, in this project, Hash Table Operations was used for storing words which comes form the Text Files that read in File Operations. There are two Hash Table Operations; Open Addresing and Chaining Method. Open Addressing includes three methods: Linear Probing, Quatratic Probing and Double Hashing. Each Methods has their own function for inserting in Hash Table. In this manner, the Text Based Search Engine project has a search operations for each Linear Probing, Quatratic Probing, Double Hashing and Chaining Method.

Finally, after searching a word/s from the documents, a similarity algorithm and sort algorithm work in background and shows us the most smiliar text file from all of the documents.

The Text Based Search Engine Project was planned seven week project for developing. In this project, Gül Eda Aydemir, Seher Balci, Ömer Beyazıt worked together.

# CHAPTER THREE

Progress Summary

The Text Based Search Engine project was planned for a seven week project for devoloping. In this project, Gül Eda Aydemir, Seher Balcı, Ömer Beyazıt worked together.

## Project-Tasks Approach

1. -Read All Documents
2. -Eliminating Stop Words & punctuation marks
3. -Frequences of Words
4. -Insert Hash Table
   * 1. -Linear Probing
     2. -Quatratic Probing
     3. -Double Hashing
     4. -Chaining
5. -Search Hash Table
6. -Similarity Algorithm
7. -Sorting Algorithm
8. -Collision & Time

## Time-Periods Approach

|  |  |
| --- | --- |
| First Week | No Lecture + Project Introduction |
| Second Week | 1,2,(3->) |
| Third Week | 3 |
| Fourth Week | (4-iv),(5->) |
| Fifth Week | 5 |
| Sixth Week | (4-i,ii,iii),8 |
| Seventh Week | 6,7 |

\*All tasks given in the Project-Tasks Approach

\*(->) means not finished, continue

# chapter fOUR

Problems Encountered & Changes In RequIrements

Throughout the Text Based Engine Project, there are some changes in requirements.

Fisrt of all, in project description paper, in first milestone, ‘printing out only the words of a particular document as the output’ was expected from our team. However, this section changes to ‘ prints out only the words of all documents as the output’ and our team didn’t know about it, so we lose points from Milestone.

Secondly, all ower, in project description paper, in second milestone, ‘storing words with using hash table with chaining method’ was expected from our team. However, this section changes to ‘ storing words with using Hash Table Chaining and Linear Probing, Quatratic Probing, Double Hashing’. Unfortunately, our team didn’t know about it, due to ground on the paper. Instead of doing ‘Linar Probing, Quatratic Probing, Double Hashing’, our team did the search algorithm. This change in requirement takes one more week. So the deadline of the project chaged to November 8th,2016 from 27th October,2016.

Finally, another change in the project description paper was about ‘collision,time and to find the effective/efficient method’. This change in requirement takes more time for our team. We have solved this problem by meeting in the weekend.

# chapter fIVE

# ClASSES

|  |  |  |
| --- | --- | --- |
| fILE OPERATION | hASHtABLE | WORDS |
| * Stopwords() * ReadFolder() * ReadFile(string file) * wordsOperation(string line) * deleteStpWrd(string str) * GetSearchedFolderName(string search) * Strin[] sort(string[]doc, string[] tempDoc) | * İnt Hash(int key,int tablesize)   // initial hash function   * İnt hashforLinear(int key, int i)   // Linear probing   * İnt hashforQuadratic(int key, int i)   // Quadratic probing   * İnt hashforDouble(int key, int i)   // Double hashing   * İnt String[] search(int key,String searchWord)   // search for chaining   * put(int key, String word, String doc)   // insert for chaining   * insert(int key, String word, String doc)   //insert for open addressing ekleme   * rehash()   // for open addressing   * String[] searchOpen(int key,String searchWord)   // search for open addressing   * String[] sortDoc(int[] fr, String[] doc) | * private String word = null; * private String doc = null; * private int fr = 0; * private int key = 0; * private Words next; * private Words down; * public Words(int key, String word, String doc, int fr) * get&set methods |

# Chapter sıx

# comparısıon hash methods

Chaining Method:

* **Formula : double num1 = 0.618033 \* key;**

**double num2 = Math.*floor(num1);***

**return (int) Math.*floor((num1 - num2) \* tableSize);***

* ***Collision : 127112***
* ***Read Time : 0.38 s*** ☺

Open Addressing:

* 1. Linear Probing:
* **Formula: return key = (key + i) % table.length;**
* **Collision: 7,057,577**
* **Read Time: 317,996s**
  1. Quadratic Probing:
* **Formula: return (int) ((key + 91 \* i + 17 \* Math.pow(i, 2)) % table.length);**
* **Collision: 657411520**
* **Read Time: ~4 hours** ☹
  1. Double Hashing:
* **Formula: int hash = (int) (((key \* 0.618) % 1) \* 1000);**

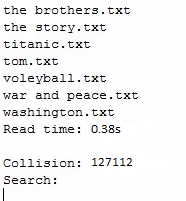
**return (key + i \* hash) % table.length**

* **Collision: 894573320**
* **Time: ~ 8 hours**☹

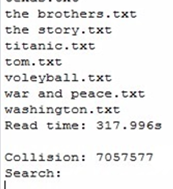
# CHAPTER sEVEN

SCREENSHOTS

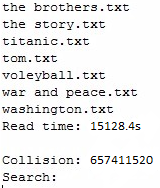
1. Hash Table Chaining Method



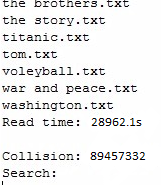
1. Hash Table Open Addressing Linear Probing



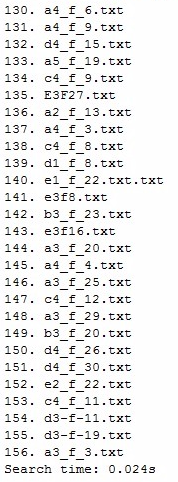
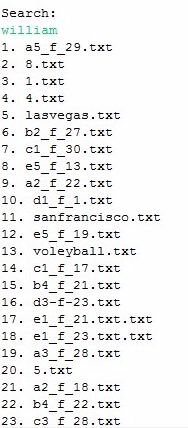
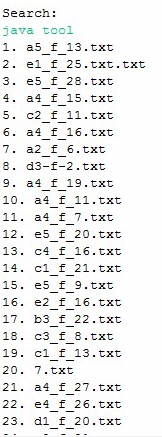
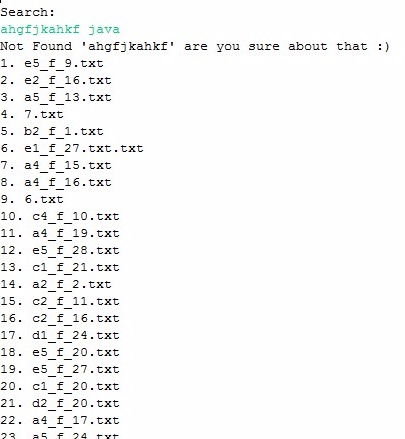
1. Hash Table Open Addressing Quadratic Probing



1. Hash Table Open Addressing Double Hashing



SCREENSHOTS FROM USER PERSPECTIVE

# CHAPTER eıght

conclusıon and future work

In conclusion, regardlessly to all changes in requirements, our Text Based Search Engine Project works very well with hash table chaining method. As shown in the ‘Comparision Hash Table Methods’, we have got a ineffectual Hash Table formula for Open Addressing Double Hashing and Quadratic Probing. Despite the fact that, chaining method insertion and searching methods works effecient and also effective.

As a future work, the Text Based Search Engine Project will be more productive with changing the Hash Table formulas. Also, can implement a smilirity method which the time that, User enteres the word wrong, but we can understand what the User means and re-showing the word to User for fast searches. Last but not least, a User account’s following a history for the searches could be thought for the future work.

Finally, with a User accounts, a huge database implementation can be a future work as well.