Hacettepe University

Computer Engineering Department

BBM 203 Software Laboratory

Fall 2016

Date Issued	**:** 12.12.2016
Date Due	**:** 26.12.2016
Advisor	**:** Dr. Ali Seydi Keçeli

Summary

In this experiment you will learn how to use binary search trees on C programming language.

Problem

In this experiment you will implement a document search application. Your application will take a directory path as a parameter then it will scan all text documents in this directory recursively and store all words in a binary search tree. You should implement your own tree structure. Do not use any third party library for tree implementation.

Your application will take two parameters. First parameters is the path of the input directory and second parameter is the command file. In command file there will be input commands to run the application. List of all command and their output format are given below. Your application should print all of its outputs to a file named output.txt and standard output. Your application should scan the input directory first. Then commands in the command file will be executed.

Input format:

binarysearcher <input_directory> <command_file>

binarysearcher ./input_dir command.txt

Commands and Output Formats:

PRINT TREE

Print all words in the tree with their frequency, depth and names of the files that contain this word.

Output Format: <COMMAND NAME>

<WORD>

<TOTAL_NUMBER>

<DEPTH>

<FILE_NAME> <NUMBER>

<FILE_NAME2> < NUMBER >

<WORD2>

<TOTAL_NUMBER>

<DEPTH>

<FILE_NAME> < NUMBER >

<FILE_NAME3> < NUMBER >

PRINT TREE ASC

Print all words in alphabetically ascending order. Output Format will be the same as the output format of PRINT TREE command.

PRINT TREE DSC

Print all words in alphabetically descending order. Output Format will be the same as the output format of PRINT TREE command.

SEARCH <WORD>

Search given word through the tree and print its frequency, depth and names of the files that contain this word.

Output Format: <COMMAND NAME>

<WORD>

<TOTAL_NUMBER>

<DEPTH>

<FILE_NAME> <NUMBER>

ADD <FILE_PATH>

Scan the given file and add all words in this file to the binary tree.

REMOVE <WORD>

Remove the given word from the tree. The node that belongs this word should be removed too.

Upper and lower case words are the same.

- "This" and "this"

Related words that are spelled differently are different.

- "Car" vs. "cars"

Sample command file:

```
PRINT TREE
SEARCH bill
REMOVE alice
PRINT TREE
ADD ./c.txt
```

Sample Output file:

```
PRINT TREE
judy
        3
        1
        a.txt 2
        b.txt 1
bill
        2
        2
        a.txt 1
        b.txt 1
mary
        2
        1
        b.txt 2
alice
        3
        5
        a.txt 3
SEARCH
bill
        2
        2
        a.txt 1
        b.txt 1
PRINT TREE
judy
        3
        1
        a.txt 2
        b.txt 1
bill
        2
        2
        a.txt 1
        b.txt 1
mary
        2
        1
        b.txt 2
```

You could perform file and directory operations using UNIX system calls. You will/may use the following system calls functions: open(), close(), lseek(), read(), fcntl(),write() stat(), fstat(), lstat(), unlink() utime(), chdir(), getcwd(), opendir(), readdir(), closedir(),rmdir()

Notes:

- Please use understandable variable and function names. Respect to C naming conventions.
- Your tree structures must be dynamic.
- Provide comment part for each of your functions. Also provide comment lines on critical parts of your codes, if necessary.
- Please strictly respect to output file formats.
- You should scan input directory and its subdirectories.
- You should check that indexed file is a text or a binary file. For binary files no operation will be handled.

Structure of you zip file

studentnumber.zip

+ src (folder)

- makefile (mandatory)

- binarysearcher.c (mandatory)

Please don't forget to rename your main file to binarysearcher.c. And there is a makefile (it is also mandatory) for you to submit.

Basically, makefile is a set of instructions for compilation of your codes. It's usually needed for large projects, but as you will see in the near future, it is very important for end users (even for you).

I share some tutorials about makefile. I want to use "make" and "make clean" commands from your makefile. I think, section 1 and 2 of 1st tutorial will be very helpful for you.

http://www.gnu.org/software/make/manual/make.html

http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/

Good luck

Remarks:

- You will use online submission system to submit your experiments.
- https://submit.cs.hacettepe.edu.tr/ Deadline is: 23:59. No other submission method (such as; CD or email) will be accepted. Late deliveries will not be accepted, either.
- Do not submit any file via e-mail related with this assignment.

- SAVE all your work until the assignment is graded.
- The assignment must be original, INDIVIDUAL work. Duplicate or very similar assignments are both going to be punished. General discussion of the problem is allowed, but DO NOT SHARE answers, algorithms or source codes.
- You can ask your questions through course's Piazza group and you are supposed to be aware of everything discussed in the group.
    - https://piazza.com/hacettepe.edu.tr/fall2016/bbm203