HACETTEPE UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

BBM-203 PROGRAMMING LAB.

ASSIGNMENT 4

Subject: Login System with Character Tree

Submission Date: 17.12.2018

Deadline: 06.01.2019 23:55

Programming Language: C Operation System: CentOS Linux

Advisors: Dr. Sevil Sen Akagündüz, Dr. Mustafa Ege, Dr. Cumhur Yiğit Özcan, R. A. Pelin Canbay

Student Name : Ali Kayadibi

Student Number: 21727432

1. Introduction / Aim

In this assignment, we are expected to design a Login System with Character Tree. The primary goal of this experiment is to get you to practice on the tree data structure.

2.Commands

Our login sistem includes five different command which are adding user,searching user,sending query with given user,deleting user and deleting all users in the tree.

-a username password #add username to the tree with the given password

-s username #search with the given username and return the password if it is

-q username password #send query for the password with the given username

-d username #delete username and its password -l #list the tree

3.User Struct

```
typedef struct Trie {

    char character;
    char password[10];
    struct Trie* childs[26];


} agac;

agac* head;
```

This is my tree struct which have character of that node,password if needed , and link to other childs.By holding character of nodes we can reach wanted node in constant time and check it’s password if needed.

4.Functions That I Used

struct Trie* nodeMaker()

My node maker function which creates the first node, fills childs of this node with null values,and returns a global pointer to trie structure.So i can use it to access other nodes of the tree.

void ekle(agac* head,char* str,char* password)

My insert function which takes head pointer of the program,string that we want to add to tree,password of the user.This function starts checking nodes if they have same character name so we can save memory by not creating new nodes.If first character of the string is not null on the head array, it checks the next character until there is no match between nodes and string.Creates it's own nodes and connects them and adds the password at the end of final node.If there is no match program just creates nodes from beginning until end of the string.Program also checks if that username is taken or not.

int search(agac* head,char* str)

Search function which takes head of the tree and the wanted person's username.Program starts looking at the node names to find wanted person.If all of the characters of the string is matched program gives password of that person.If even the first character of the string is not in tree program gives no record error , if some of the characters of given string exist program gives not enough username .If only first n character of the string exist program gives incorrect username as a eror.Program returns different integer value for possible error occurs.

void searchOut(int x)

Selects one of the cases with the given integer value and prints that case's error.

void query(agac* head,char* str,char* password)

Query function which takes head of the tree,username of the person who wants to enter the program,password of that person.First program searchs for the given user.If user is not found program gives no record error.If some of the chars of the user is found program gives incorrect username or not enough username if there is no password of that user.If user is found program checks if password of user is same with the value in the tree or not.If not program gives incorrect password if passwords match program gives succesful login as a output.

void delete(agac* head,char* str)

Detele function which takes head of the tree and the username that we want to remove.First program searchs if there is a user with that username.If there is no username program gives no record error.If first character of ther user exist but not others program gives incorrect username, if there is no password with wanted username program gives not enough username error as a output.If program finds given user it starts deleting it.There are multiple cases.It may not delete it because other user is using same nodes, there might be only 1 user using these nodes or person might be subtree of another user.We check these conditions and after that program deletes these nodes with wanted action.

void display(agac* root, char str[], int level,int dal)

My print function which prints takes head of the tree,one string array,avaible index in the tree,and if there is a dallanma in the tree or not.It prints common characters of the users and than prints their full names.It is a recursive function which looks if there is a avaible child in the tree, if there is function calls itself with address of that child and it calls until there is no node in the tree.In every call function checks three conditions. Which are it can be the last node on the list so we will print it, it may not be last but it may have a password we will print it, it may be the common characters of these names program will print it, it may be single name with no childs in every node and program will print it too.

5.My Detailed Algorithm

Program starts with taking single command line arguments which is name of the input file.Program creates a struct which have character,password and link to other child attributes.After creating head of the tree, I start reading input file and use them in my functions.If the input is insert command program takes name and password of the given user and calls insert function and inserts them into the tree.In insertation program checks if characters of the tree is already created or not.Purpose of doing this is to use lesser memory usage and we can access these nodes in much faster time.If first n character of the string is already exist, program uses these previous nodes and links the new nodes in these nodes.If there is no common character, program starts creating new nodes starting from the first child of head until last child of it.And adds password to last node.If the wanted command is search program starts looking at the node names to find wanted person.If all of the characters of the string is matched program gives password of that person.If even the first character of the string is not in tree program gives no record error , if some of the characters of given string exist program gives not enough username .If only first n character of the string exist program gives incorrect username as a eror.Program returns different integer value for possible error occurs.If the wanted command is query command which is similar to login system. Program searchs for the given user.If user is not found program gives no record error.If some of the chars of the user is found program gives incorrect username or not enough username if there is no password of that user.If user is found program checks if password of user is same with the value in the tree or not.If not program gives incorrect password if passwords match program gives succesful login as a output.If the wanted command is delete, it deletes if program finds the user. First program searchs if there is a user with that username.If there is no username program gives no record error.If first character of ther user exist but not others program gives incorrect username, if there is no password with wanted username program gives not enough username error as a output.If program finds given user it starts deleting it.There are multiple cases.It may not delete it because other user is using same nodes, there might be only 1 user using these nodes or person might be subtree of another user.We check these conditions and after that program deletes these nodes with wanted action.And the last command is list command

which is to print all the names on the tree.It also prints first n common character of the users.This is all command that my program supports.It also uses trie implementation instead of writing it with linked list because this way it is much faster.