PROGRAMMING ASSIGNMENT 4

21604552 - Emre Hancı

Hacettepe University - Department of Computer Engineering
BBM203 Software Lab.

-What was the problem?

In this assignment our advisors expect that to teach us “How can I use Tree struct?” and “How can I use Trie struct?” with a simple implementation about natural language processing. This program is about storing username and password on trie implementation. While doing this process there are several command which are reading and processing on input file;

1)With -a command:

*If given username with -a command is already used for anyone else, program should print-out “Reserved Username”

*If given username with -a command have not been used for anyone else, program should add this username and password to the tree.

2)With -s command:

*If given usernames not even one char is on the tree program should print-out “No Record.”

*If given usernames first char is on the tree but remainder not, then program should print-out “Incorrect Username”

*If all the char is on the tree but last char of username has not a password(In my program that’s mean password is equal to ”/”) program should print-out “not enough username”

*If given username is exist on tree then program should print-out the password of given username with “password xxx” format

3)With -q command:

*If given usernames first char is not a child of root node then program should print-out “No record”

*If given usernames first char is a child of root node but remainder not then program should print-out “Incorrect Username”

*If all the char is on the tree but last char has no password then program should print-out “Not enough username”

*If all the char of username is on the tree but given password is not same with storing password then program should print-out “Incorrect password”

*If given username is exist and given password is same with storing password then program should print-out “Successful Login”

4)With -d command:

*If given usernames first char is not on the tree then program should print-out “No record”

*If given usernames n char is on the tree but remainder is not then program should print-out “Incorrect Username”

*If given usernames all char is on the tree but that username is not taken by anybody then program should print-out “Not enough username”

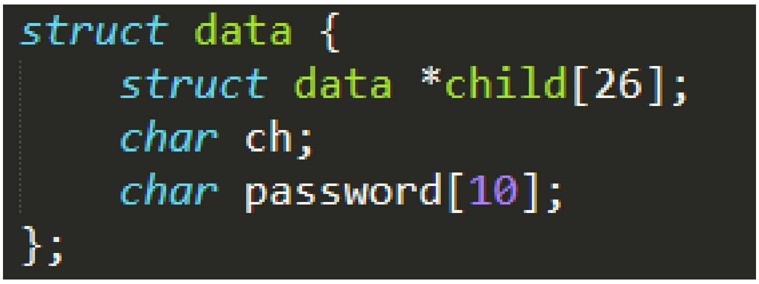
*If given usernames all char is on the tree and last char has password then program should remove necessary nodes on the tree then print-out “Deletion is succesful”

5)With -l command

*When this command given, program should print-out the tree.

-The Data Structs I Used;

Data;



```
struct data {
    struct data *child[26];
    char ch;
    char password[10];
};
```

I create this struct for storing the tree information about users. I use a character for being sure that root is empty, that char is “/”. When a node is empty that nodes ch and password is equal to “/”.

-The explanation of my functions

Functions for output file;

*void printToFile(char *line)*

This function for printing given char array to output.txt

void printToFileCh(char line)

This function for printing given char to output.txt

Functions for processing command on input file;

*void init(struct data *Root)*

This function for initializing the root element.

*struct data *createNode()*

This function creates new node and returns a pointer to point it.

*struct data *search(struct data *Root,char *Name)*

This function for search given name in the tree.

*int append(struct data *Root,char *Name,char *Password)*

This function for append new user to tree.

*int login(struct data *Root,char *Name,char *Password)*

This function for checking login according to username and password.

*int hasChild(struct data *Root)*

This function returns 1 if given node has least one child.

int hasTwoChild(struct data Root)*

This function returns the number of child that given node has.

int returnChild(struct data Root)*

This function returns the first child of given node.

*void removeNode(struct data *Root,char *Name)*

This function removes given name if exist.

*int delete(struct data *Root,char *Name)*

This function checking deleting condition.

Functions which are working for listing tree;

*int list(struct data *Root)*

This function prepare the first printing names comes from main root.

void listRec(struct data Root)*

This function prepare the first printing names comes from main root.

void tabNames(struct data Root)*

This function make a direction from given node to other nodes that have password.

void tabNamesRec(struct data Root, int index)*

This function is a rec. function which is prepare the names after the first collison from main root.