Hacettepe University
Computer Science and Engineering Department
Name and Surname : Burak Karademir
Identity Number : 21527123
Course : BBM 203
Experiment : ASSIGNMENT 4
Subject : Login System with Character Tree
Due Date : 06.01.2019 23:55
Advisors :Dr. Sevil Sen Akagündüz, Dr. Mustafa Ege, Dr. Cumhur Yiğit Özcan, R. A. Pelin Canbay
E-mail : karademirburak@outlook.com
Programming Language : C

2. Software Using Documentation

2.1. Software Usage

I used netbeans for this assignment.My assignment takes 2 command line arguments. One for input file and one for output file name. I use dynamic memory allocation. We have one input files and one output file.My program reads input files and writes output to a text file which name is given with command line arguments. Also input file names are given by command line arguments.

3.Software Design Notes

3.1. Description of the program

3.1.1. Problem

We have one input files.Input file includes commands about login system(For example -a adds the username and password, -d deletes the username and password). Firstly we need to create a tree and read the input file.Then we need to apply the commands which are in the input file. Then we need to write functions for applying the commands .First command is -a command, this command adds the username and password. Second command is the -s command, this command searches the given username. Third command is -q command, this command read the username and its password and according to the matching, it will return an information. Fourth command is -d command, this command deletes the username and password. Final command is -l command, this command lists the usernames.

3.2. System Chart

INPUT	PROGRAMS	OUTPUT
input.txt (given by command line)	main.c	output.txt(given by command line)

Name of the input and output file can change since we take the name of the inputs and output from the command line.

3.3. Main Data Structures

I used tree data structure .I also used dynamic memory allocation for chars and tree struct.

3.4. Algorithm

In the main firstly I open a file for writing outputs which name is taken by command line arguments.(argv[2]). Then I find the input file's line number using find_line_number function. Then I use dynamic memory allocation for struct lines(it keeps the input file lines). After that I call read_file function for reading the input file. Then I make a declaration (int i) for using it in loops.Then I create root node. After that I write a loop for commands and I check all commands with if else statements. In the for loop firstly I use strtok for seperating the lines(I use space and newline character as a delimiter). After that firstly I check -a command for adding the username and password to tree. In -a command firstly I search the username and if it exists then my program writes "reserved username" and if it is not exist in tree then I insert it and my program writes "..... was added". After -a command I check -s command. In -s command firstly I take the username's first character. Then I check the username and if it exist in tree and it has password then my program writes "password xxx".(I use searchpass and printpass functions for this).Then I check if all characters of the username exist on the tree, but the last character has no password, the application will give an output that "not enough username". (I use search_withoutendofword function for this). Then I check if the first n character of the username exists on the tree, but the remainder is not, the application will give an output that "incorrect username".
(I use search_firstchar function for this).

After that I check if the first character of the username is not referenced by the root node the application will give an output that “no record”. (I use search_firstchar function for this). After that I check -q command. In this command I check if all characters of the username exist on the tree, and the last character has the same password with the given, the application will give an output that “successful login”.(I use searchand_comparepass function for this). I use same function for other situations (like incorrect password) . After -q command I check the -d command. In this command firstly I check if all characters of the username exist on the tree, and the last character has the password, the application will delete all nodes which are not connected to another username. Then it will give an output that “deletion is successful”.(I use deletion function for this). I use same function for other situations. Finally I check -l command and I list all usernames with display method. In this method, I make each branch after the first level of the tree printed as a root. And usernames starts with that root are printed under that root.