Hacettepe University
Computer Science and Engineering Department
Name and Surname : İdil CAN
Identity Number : 21727011
Course : BBM203
Experiment : 2018 Autumn Semester Programming Assignment 4
Subject : Login System with Character Tree
Data Due : 6th January 2019
Advisors : Sevil Sen Akagündüz, Mustafa Ege, Cumhur Yiğit Özcan, Pelin Canbay
e-mail : idil.can@hacettepe.edu.tr

INDEX OF THE REPORT

1. SOFTWARE DESİGN NOTES

1.1. Description of the program

In this assignment we were expected to design a login system using trees. There is five different commands that the user may give: add, delete, query, search and list. The command add(-a) and query(-q) takes two values: name and password. Delete(-d) and search(-s), on the other hand, takes only the name. Apart from all these, list(-l) does not take any other inputs and lists all the users.

1.2. Algorithm

The tree implementation that has been used in this assignment is really similar to linked list. Nodes have 3 different qualities: character, next nodes' addresses and a password. All those qualities are assigned to "NULL" while initializing the nodes.

The nature of this data structure makes the usage of recursion, almost, compulsory. Without recursion the algorithm becomes really complicated and harder to read, if we assume that we managed to write it correctly. So, for almost every command we had to use recursion and I think this assignment improved my ability to think recursive.

For every function except "-l", I used the same searching method with minor changes. For "-a" function I look for the elements of the base node (starting from ROOT), then in the point that there is no path to follow I start to create the new path. But if there is no path to create, I check the password. If it's null, I initialize the password. If it's been initialized before I return a error message.

For "-s" function, I basically search through the array of next nodes' addresses. At the point that there is no initialized node at where should the nest character be, it returns an error display. If the elements of the name are finished but there is no password attached, it still returns an error message. Only when I find the name and the password attached to it, it returns the password.

When user gives the “-d” function, it searches through the tree and finds the right leaf of the tree. Every node returns a value that informs the upper node about its condition: can be deleted or not. If it can be deleted, it frees the memory location and assigns the pointer to “NULL”. If not, returns the condition.