Mert Kılıçaslan

File Systems (PA – 5) Report

CS307 – Operating Systems

Description of the Algorithms to solve the problem:

My algorithm starts with opening the root database.txt file to store person attributes to the array of struct for avoiding complexity during the file handling. After the successful opening and storing process, the algorithm passes the array, its size, and the root directory as an argument to the recursive 'listFiles' function which calls ‘fixFiles’ function during its recursion.

The listFiles function creates an entity object with the directory pointer to recursively iterate until there are no more directories left which means it recursively scans until only files with extensions remain such as ‘.txt’, ‘.dat’. It also concatenates the path of the directory after each step of going down. For example ./a, ./a/b, ./a/b/.test.txt. After the recursions are completed, an if condition checks whether the files left have the ‘.txt’ extensions or not. If it is, then it calls fixFiles function with an argument of the path that corrects the misspelling words later on with the use of fseek and fputs.

Similar to the listFiles function, the struct and its size were passed as an argument for the fixFiles function. However differently from it, this time there is a directory name to find the locations of the text files to be fixed. The fixFiles function is being called for every path passed as an argument except the root ./database.txt which plays the main resource role for us even we had stored the attributes in a struct before. Therefore the function skips the root database.txt. For the other paths, it opens in ‘r+’ mode to scan each word one by one and overwrite afterward. If the word exists in the struct as a name attribute; with the use of fseek,

then the file pointer goes back for the value of the current location minus length of the name and minus 4 for the title. Then it looks if the person’s name is a female name or male name after it decides, it concertantes according to the title, the name, and the last name to overwrite the whole phrase with fputs function. After all, names are corrected, the file is closed and the function finishes its execution and will be called for the next path.