Programming Assignment 4 Report

Hüseyin Alper Karadeniz

Operating Systems (CS 307) / Fall 2022-2023

1 Process and Program Flow

I used C programming language for the implementation of this homework. This program corrects all of the text files in which there are incorrectly written person information by the last names or titles, according to the given database in the beginning. My program performs necessary operations in the text files in the root directory, and in the sub-directories of it by reading `database.txt` file in the root directory, which stores the information of people in our database.

I have declared four global variables in my program, and I initialized two variables. I used these declarations for the ease of use during the program, and I used the initializations to maintain the functionality of the program.

In the program, I have created a structure called `Person`, which has attributes for the name, surname, and gender information of the person. People are identified by their names, and it is assumed that each name is unique so that there won't be any overlaps in the iterations of the people. Any requested person can be found in the database from the name with `findIndex()` function.

1.1 main()

First of all, my program reads `database.txt` file in the root directory line by line and it stores all the data (gender, name, surname) of the people there into an array called *people*, which is a global variable. I used `fscanf()` function in order to split the data in each line there, it reads all three strings in the line. Then, it saves the information by creating a `Person` with them. After creating *people* array, the program closes the text file storing the database with `fclose()` function. Then, the program starts a recursive file search inside of the directories with `modifyDirectory()` function.

1.2 modifyDirectory()

The program opens the desired directory with `opendir()` function, and it reads the contents in the directory by traversing directories one by one with

`readdir()` function. It stores the path for each entry. If the entry is a sub-directory (DT_DIR), then a recursive `modifyDirectory()` function is called for the sub-directory. Else, if the entry is a regular file (DT_REG), then the file format of the entry is checked if it is a text (.txt) file, or not. If the entry is a text file other than `database.txt` file in the root directory, then `modifyFile()` function is called for the path of the entry. When all operations are done in the directory, it is closed with `closedir()` function.

1.3 modifyFile()

The program opens the desired text file in $r+$ mode with `fopen()` function. After that, before the iterations, some variables are declared to store the person data in the input file, which will then compared with the original ones. After these declarations, the program starts to read the text file word by word with `fscanf()` function. If the program finds a string which is "Ms." or "Mr." meaning that we found a person, it gets two more strings so that gets the name and the surname of the person. We had an array called *people* storing the data of the people in the database. The program uses this array and finds the original (correct) gender and surname of the given name. Thus, now we have both the current data of the person in the text file, and the correct version of the data. We will update the information of the person on the text file if there are any mistakes. The program finds the index where it need to start change operations by using `fseek()` function. After finding the correct index, it replaces the correct information with the incorrect information on the current text file with the help of `fputs()` function. When all operations are done in the text file, it is closed with `fclose()` function.

2 Summary

This program modifies text files by changing incorrectly written titles and surnames of the people from their names in the database. For the database, there is an input file having name, surname, and gender information for each person. The program recursively modifies all of the files in a directory, including all the files in its all sub-directories according to the given database input file.