МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №3
по дисциплине «Программирование»
Тема: Строки. Рекурсия, циклы, обход дерева

Студент гр. 3341 _____ Романов А. К.

Преподаватель _____ Глазунов С.А.

Санкт-Петербург
2024

Цель работы

Цель работы заключается в разработке программы на языке программирования, которая осуществляет рекурсивный обход иерархии папок и файлов в заданной структуре, анализирует названия текстовых файлов, записывает их полные пути в виде строки в файл.

Задание

Вариант 4

Дана некоторая корневая директория, в которой может находиться некоторое количество папок, в том числе вложенных. В этих папках хранятся некоторые текстовые файлы, имеющие имя вида <filename>.txt. В качестве имени файла используется символ латинского алфавита.

На вход программе подается строка. Требуется найти и вывести последовательность полных путей файлов, имена которых образуют эту строку.

При этом учесть:

- Регистрозависимость
- Могут встречаться файлы, в имени которых есть несколько букв и эти файлы использовать нельзя.
- Одна буква может встречаться один раз.

Ваше решение должно находиться в директории /home/box, файл с решением должен называться solution.c. Результат работы программы должен быть записан в файл result.txt. Ваша программа должна обрабатывать директорию, которая называется tmp.

Выполнение работы

Используемые переменные:

- *char string[100]* — используется для хранениея искомого слова. Вводится пользователем
- *int length* — хранит информацию о длине введенной строки string
- *const char *directory* — содержит адрес директории, в которой требуется анализировать файлы. (В условиях данного варианта - ./tmp)
- *char path[]* - содержит путь файла, в который будет записываться ответ. (В условиях данной работы - ./result.txt)
- *FILE *result* — собственно файл, в который записывается ответ.

Реализованные функции:

- *void match(const char *dirPath, FILE *result, char str)* — принимает на вход адрес обрабатываемой директории, файл, в который будет вестись запись ответа, а также символ *str*. (В последствии функция сравнивает названия файлов с этим символом). Функция обходит все файлы в указанной директории. В случае если встречается директория, функция вызывает саму себя, передавая в качестве параметра *dirPath* адрес найденной директории. В случае если очередной файл не является директорией, функция проверяет, что его имя состоит из одного символа, и если символ совпадает с *str*, осуществляется запись полного пути найденного файла в .txt документ *result*.
- *int main()* - объявляет вышеуказанные переменные. После чего перебирает символы в строке *string* и для каждого символа вызвает функцию *match*.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Введенная строка: HeLlO	hello_world_test/asdfgh/mkoipu/H.txt hello_world_test/qwerty/e.txt hello_world_test/qwerty/qwert/L.txt hello_world_test/asdfgh/l.txt hello_world_test/asdfgh/O.txt	Тест с e.moevm
2.	Введенная строка: FOrTnitE	/home/alex/Desktop/ffff/F.txt /home/alex/Desktop/ffff/folder/O.txt /home/alex/Desktop/ffff/folder/r.txt /home/alex/Desktop/ffff/folder pro/T.txt /home/alex/Desktop/ffff/folder pro/n.txt /home/alex/Desktop/ffff/i.txt /home/alex/Desktop/ffff/t.txt /home/alex/Desktop/ffff/folder/E.txt	Тест на моей системе… (проверял на этом работоспособность программы)
3.	Введенная строка: word	/home/alex/Desktop/ffff/folder/r.txt	Та же директория. Программа нашла только один файл, имя которого присутствует в строке, что соответствует действительности.

Выводы

В ходе выполнения данной работы были приобретены навыки эффективного использования рекурсивных методов для обхода дерева файлов, а также работы с файловой системой, анализа данных о файлах и записью информации в файл. Разработка программы, способной автоматически обрабатывать информацию из различных файлов и директорий, позволила улучшить навыки программирования и решения сложных задач.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: solution.c

```
#include <stdio.h>
#include <dirent.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <string.h>

#define FILEPATH_BUFFER 1024
#define STRING_BUFFER 100
#define SEARCH_DIRECTORY "./tmp"
#define ANSWER_FILE "./result.txt"


void match(const char *dirPath, FILE *result, char str);

int main()
{
    char string[STRING_BUFFER];
    scanf("%s", string);
    const char *directory = SEARCH_DIRECTORY;
    int length = strlen(string);

    char path[] = ANSWER_FILE;
    FILE *result = fopen(path, "w");
    result = freopen(path, "a", result);

    for(int i =  0; i < length; i++)
    {
        match(directory, result, string[i]);
    }
    fclose(result);

}

void match(const char *dirPath, FILE* result, char str)
{
    if(result)
    {
        DIR *dir = opendir(dirPath);

        if(dir)
        {
            struct  dirent *file_in_dir = readdir(dir);

            while(file_in_dir)
            {
                struct stat statbuf;
                char filepath[FILEPATH_BUFFER]="";
                strcat(filepath, dirPath);
                strcat(filepath, "/");
                strcat(filepath, file_in_dir->d_name);
```

```
                    if(file_in_dir->d_name[0]!='.')
                    {
                        if(stat(filepath, &statbuf)==0)
                        {
                            if(S_ISDIR(statbuf.st_mode))
                            {
                                match(filepath, result, str);
                            }
                        }
                    }

                    if(file_in_dir->d_name[0]  ==  str  &&  file_in_dir-
>d_name[1] == '.')
                    {
                        fprintf(result,"%s/%s\n", dirPath, file_in_dir-
>d_name);
                    }
                    file_in_dir = readdir(dir);
                }
            }
            closedir(dir);
        }
    }
```