**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: Обход файловой системы**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 3342 |  | Смирнова Е.С. |
| Преподаватель |  | Глазунов С.А. |

Санкт-Петербург

2024

## **Цель работы**

Целью работы является освоение работы с рекурсивными функциями и файловой системой, а также ее рекурсивным обходом.

## **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.

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**

Для выполнения поставленной задачи была реализована рекурсивная фукнция void serch\_letter(char letter, char \*file\_path, FILE\* file).

В качетсве первого аргумента, в функцию передается букву, которую нужно найти в данный момент. Далее в функцию передается адрес той директории, в которой будет осуществляться дальнейший поиск, а в качестве последнего аргумента в функцию передается файл, в который нужно осуществить запись полученного результата.

В начале функции мы получаем доступ к нужной нам директории и осуществляем проверку. Если директория открылась успешно, получаем очередной элемент открытой директории. Далее с помощью условия if-else определяем, какую инструкцию необходимо выполнить. Если объект является директорией, то мы формируем путь, включая туда эту директорию и заново вызываем функцию уже для нового сформированного пути. Если объект является файлом, мы проверяем, сколько букв стоит в названии файла и является ли буква подходящей. Если эти условия выполнены, осуществляется запись в файл.

В функции main() осуществляется вызов фукнции search\_letter() в цикле for.

Разработанный программный код см. в приложении А.

## **Тестирование**

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |
| --- | --- | --- |
| № п/п | Входные данные | Выходные данные |
| 1. | 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 |

## **Выводы**

Были изучены методы взаимодействия с файловой системой в языке Си. Также была разработана программа, позволяющая записывать в файл последовательность полных путей файлов, имена которых составляют введенную строку.

# **ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ**

Название файла: main.c

#include <stdio.h>

#include <string.h>

#include <dirent.h>

void search\_letter(char letter, char\* file\_path, FILE\* file) {

DIR\* dir = opendir(file\_path);

if (dir) {

struct dirent\* de = readdir(dir);

while (de) {

if ((de->d\_type == DT\_DIR) && (strcmp(de->d\_name, ".") != 0) && (strcmp(de->d\_name, "..") != 0)) {

char\* new\_file\_path[strlen(file\_path) + strlen(de->d\_name) + 1];

sprintf(new\_file\_path, "%s/%s", file\_path, de->d\_name);

search\_letter(letter, new\_file\_path, file);

}

else {

if ((strlen(de->d\_name) == 5) && (de->d\_name[0] == letter))

fprintf(file, "%s/%s\n", file\_path, de->d\_name);

}

de = readdir(dir);

}

}

closedir(dir);

}

int main() {

char word[100];

scanf("%s", word);

char\* file\_path = "./tmp";

FILE\* file = fopen("result.txt", "w");

for (int i = 0; i < strlen(word); i++)

search\_letter(word[i], file\_path, file);

fclose(file);

return 0;

}