**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:

* *сhar 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 |
|  | 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 |
|  | 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) |
|  | 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);

}

}