**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: РЕКУРСИЯ, ЦИКЛЫ, РЕКУРСИВНЫЙ ОБХОД

ФАЙЛОВОЙ СИСТЕМЫ

|  |  |  |
| --- | --- | --- |
| Студент гр. 3341 |  | Трофимов В.О. |
| 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.

Для достижения поставленной цели требуется решить следующие задачи:

1) Ознакомиться с понятием рекурсии;

2) Освоить написание рекурсивных функций в языке Си;

3) Изучить работу с файловой системой в языке Си;

4) Написать программу для рекурсивного обхода всех файлов в папке в

том числе во вложенных папках.

## Задание

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.

Входная строка:

HeLlO

Правильный ответ:

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

! 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.

## Основные теоретические положения

1. Работа с файловой иерархией - это процесс управления структурой файлов и директорий на диске компьютера. Файловая иерархия представляет собой иерархическую структуру, состоящую из файлов и директорий, которые могут содержать другие файлы и директории.

2. Работа с файлами и директориями включает в себя создание, удаление, копирование, перемещение и изменение файлов и директорий. Операции с файлами могут включать чтение данных из файла, запись данных в файл, проверку существования файла, получение метаданных файла и другие манипуляции. В языке программирования C основные функции для работы с файлами включают в себя: fopen() - функция открывает файл и возвращает указатель на его начало для дальнейшей работы; fclose() - функция закрывает файл, освобождая его ресурсы; fread() - функция считывает данные из файла; fwrite() - функция записывает данные в файл; fseek() - функция перемещает указатель в файле на заданную позицию; ftell() - функция возвращает текущее положение указателя в файле; feof() - функция проверяет, достигли ли мы конца файла; remove() - функция удаляет файл; rename() - функция переименовывает файл; rewind() - функция перемещает указатель в начало файла.

3. Рекурсия - это метод, при котором функция вызывает саму себя в своем теле. Рекурсия позволяет решать задачи, которые могут быть разделены на подзадачи той же природы. Рекурсивные алгоритмы используются, например, для обхода файловой системы, сортировки данных, вычисления факториала и других задач.

4. Рекурсивный обход директорий - это способ обработки всех файлов и директорий внутри заданного каталога с использованием рекурсии. При рекурсивном обходе директорий функция вызывается сама для каждой поддиректории, позволяя обойти все уровни вложенности файловой системы и выполнить необходимую обработку файлов и директорий. Рекурсивный обход директорий позволяет эффективно обрабатывать большие объемы данных, которые хранятся в файловой системе.

## Выполнение работы

В программе объявлены следующие функции:

1) int validator(const char\* str, const char\* file\_name, int count);

2) int not\_cur\_parent(char\* d\_name);

3) char\* pathcat(const char\* dir\_name, const char\* file\_name);

4) void list\_dir(const char\* dir\_name, const char\* str, int index, FILE\* file);

1. Функция validator проверяет, что название файла состоит из одной латинского буквы, и латинская буква находиться в составе строки, которая подавалась на вход.

2. Функция not\_cur\_parent проверяет, что имя директории не является текущей "." или родительской "..". Если это так, функция возвращает 1, иначе 0.

3. Функция pathcat создает новый путь, объединяя имя директории и имя файла.

4. Функция list\_dir открывает указанную директорию, перебирает все элементы в ней. Если элемент является файлом и его имя соответствует validator, то записывает путь к этому файлу в файл "result.txt". Если элемент является директорией и удовлетворяет условию not\_cur\_parent, то вызывает рекурсивно list\_dir для этой директории.

В функции main, программа считывает введенную пользователем строку, открывает файл "result.txt" для записи и вызывает функцию list\_dir для каждого символа из строки, передавая путь к директории "./tmp", введенную строку, индекс символа и файл для записи результатов.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
|  | HeLlO | ./tmp/asdfgh/mkoipu/H.txt  ./tmp/qwerty/e.txt  ./tmp/qwerty/qwert/L.txt  ./tmp/asdfgh/l.txt  ./tmp/asdfgh/O.txt7 | Тест с moevm, прошёл успешно |
|  | HHOOUUEE | ./tmp/H.txt  ./tmp/H.txt  ./tmp/O.txt  ./tmp/O.txt  ./tmp/U.txt  ./tmp/U.txt | ./tmp/H.txt  ./tmp/H.txt  ./tmp/O.txt  ./tmp/O.txt  ./tmp/U.txt  ./tmp/U.txt |
|  |  |  | При пустом тесте всё нормально завершается. |
|  | CTERSLFSADF |  | Не нашлось файлов содержищие буквы строки. |

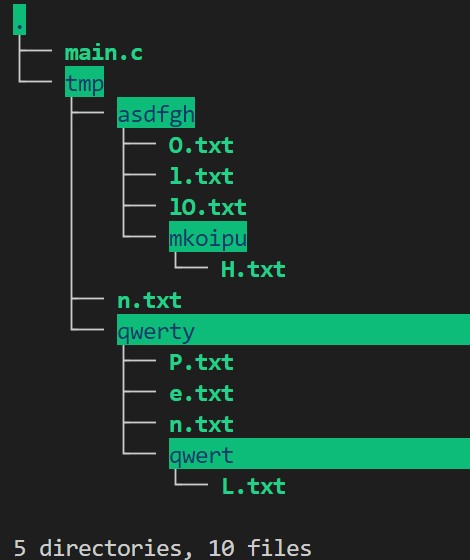


Рисунок 1- Файловое дерево для тестирования программы



Рисунок 2 – Файловое дерево для тестирования собственных тестов

## Выводы

Цель работы была достигнута, освоена работа с рекурсивными функциями и файловой системой, а также ее рекурсивным обходом. Изучены операции, используемы при работе с файлами. Реализована программа, которая рекурсивно обходит все файлы в указанной папке, включая файлы во всех вложенных подпапках, которая выбирает файлы, состоящие из одного латинского символа.

# Приложение А Исходный код программы

Название файла: main.c

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <dirent.h>

#define FILENAMESIZE 5

#define CURRENT\_DIR "."

#define PARENT\_DIR ".."

int validator(const char\* str, const char\* file\_name, int count){

int flag = 0;

if (strlen(file\_name) == FILENAMESIZE && str[count] == file\_name[0]){

flag++;

}

return flag;

}

int not\_cur\_parent(char\* d\_name){

if (strcmp(d\_name, CURRENT\_DIR) != 0 && strcmp(d\_name, PARENT\_DIR) != 0){

return 1;

}

else return 0;

}

char\* pathcat(const char\* dir\_name, const char\* file\_name){

char\* full\_path = (char\*) calloc(strlen(dir\_name) + strlen(file\_name) + 2, sizeof(char));

sprintf(full\_path,"%s/%s", dir\_name, file\_name);

return full\_path;

}

void list\_dir(const char\* dir\_name, const char\* str, int index, FILE\* file){

DIR\* dir = opendir(dir\_name);

if (dir){

struct dirent\* dir\_element = readdir(dir);

while (dir\_element) {

if (dir\_element->d\_type == DT\_REG && strstr(dir\_element->d\_name,".txt")){

if (validator(str, dir\_element->d\_name, index)){

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

}

}

else if (dir\_element->d\_type == DT\_DIR && not\_cur\_parent(dir\_element->d\_name)){

char\* new\_dir = pathcat(dir\_name, dir\_element->d\_name);

list\_dir(new\_dir, str, index, file);

}

dir\_element = readdir(dir);

}

closedir(dir);

}

else

printf("Can't open the directory\n");

}

int main() {

char string[1000];

const char\* dir = "./tmp";

fgets(string,1000,stdin);

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

result\_file = fopen("result.txt","w");

for (int i = 0; i < strlen(string); i++){

list\_dir(dir, string, i, result\_file);

}

fclose(result\_file);

return 0;

}