

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:

HeLIO

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 латинской буквы, и латинская буква находится 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
1.	HeLIO	./tmp/asdfgh/mkoipu/H.txt ./tmp/qwerty/e.txt ./tmp/qwerty/qwert/L.txt ./tmp/asdfgh/l.txt ./tmp/asdfgh/O.txt7	Тест с моеvm, прошёл успешно
2.	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
3.			При пустом тесте всё нормально завершается.
4.	CTERSLFSADF		Не нашлось файлов содержащие буквы строки.

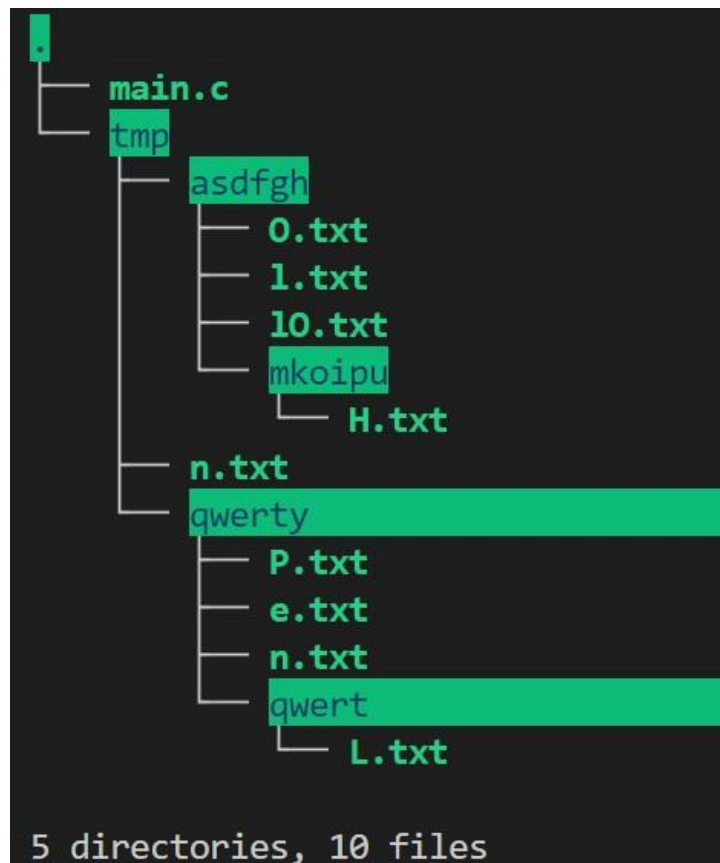


Рисунок 1- Файловое дерево для тестирования программы

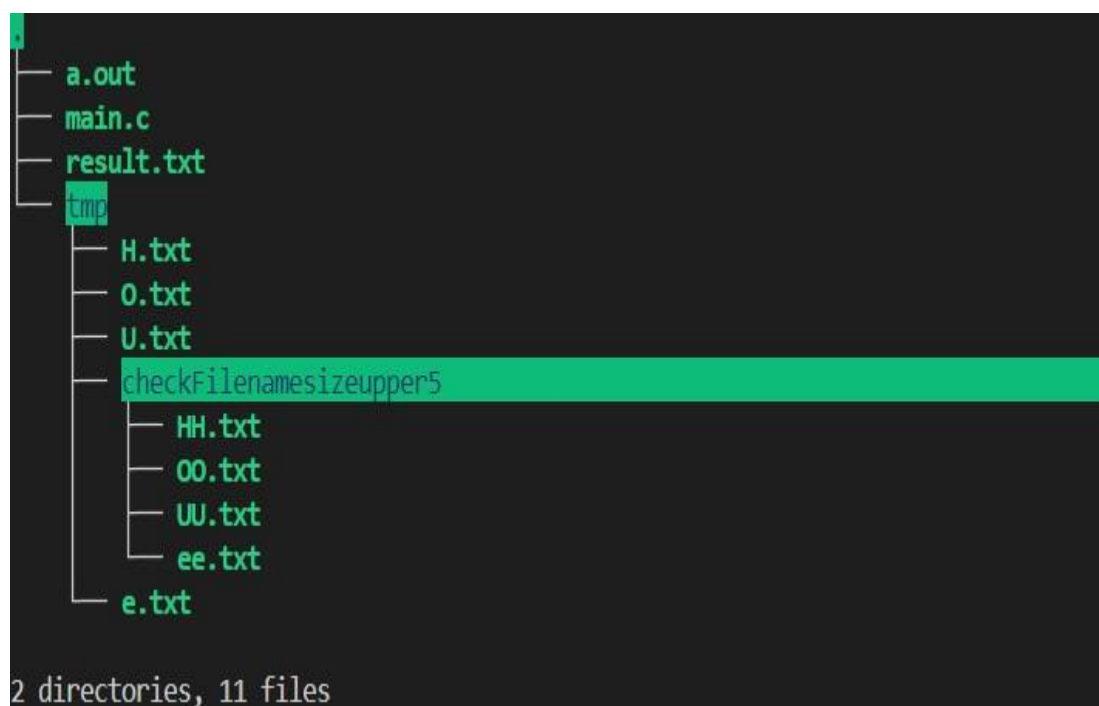


Рисунок 2 – Файловое дерево для тестирования собственных тестов

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;
}

```