

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. 3344

Бубякина Ю.В.

Преподаватель

Глазунов С.А.

Санкт-Петербург

2024

Цель работы

Освоение работы с рекурсией на языке Си на примере использующей ее программы.

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.

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.

Выполнение работы


В функции main происходит открытие/создание файла result.txt, в который будет записываться результат работы программы. После этого на вход принимается входная строка. По символам этой строки начинается обход с помощью цикла. Для каждого символа вызывается функция Files. Эта функция принимает на вход файл, путь и символ. Она записывает в переменную path обновлённый путь, если следом идущий элемент директория, иначе проверяет название файла и в случае, если он удовлетворяет условию, записывает в файл. С обновлённым путём она снова запускается. В конце работы происходит закрытие директории.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	 hello_world_test ~/Рабочий стол/hello_world_test └─ asdfgh └─ mkoipu ├─ H.txt ├─ HH.txt ├─ L.txt ├─ LO.txt ├─ O.txt └─ qwerty └─ qwert ├─ L.txt ├─ e.txt ├─ P.txt └─ n.txt	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

Была освоена работа с рекурсивными функциями на языке Си на примере использующей их программы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: solution.c

```
#include <dirent.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void Files(FILE *inp, char *main_path, char letter)
{
    DIR *dir = opendir(main_path);

    if (!dir)
        return;

    struct dirent *dp = readdir(dir);
    char *path;

    while (dp != NULL)
    {
        if (dp->d_type == DT_DIR && strcmp(dp->d_name, ".") != 0 &&
            strcmp(dp->d_name, "..") != 0)
        {
            path = (char *)malloc(sizeof(char) * (strlen(main_path) +
            strlen(dp->d_name) + 2));
            path[0] = '\\0';

            strcat(path, main_path);
            strcat(path, "/");
            strcat(path, dp->d_name);
            Files(inp, path, letter);

            free(path);
        }
        else
        {
            if (dp->d_type == DT_REG && strlen(dp->d_name) == 5 && dp-
            >d_name[0] == letter && dp->d_name[1] == '.' &&
                dp->d_name[2] == 't' && dp->d_name[3] == 'x' && dp-
            >d_name[4] == 't')
            {
                fprintf(inp, "%s/%s\\n", main_path, dp->d_name);
            }
        }
        dp = readdir(dir);
    }
}
```

```

        closedir(dir);
    }

int main()
{
    char *main_path = "./tmp";
    FILE *file = fopen("./result.txt", "w");

    char *inp_string = (char *)malloc(sizeof(char) * 1000);
    scanf("%s", inp_string);

    for (size_t i = 0; i < strlen(inp_string); i++)
        Files(file, main_path, inp_string[i]);

    fclose(file);
    free(inp_string);

    return 0;
}

```