

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №7
по дисциплине «Программирование»
ТЕМА: РЕКУРСИВНЫЙ ОБХОД ДЕРЕВА ФАЙЛОВ

Студент гр. 6303

Курков Д. В.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2017

Цель. Научится пользоваться средствами языка Си для работы с файловой системой компьютера, на практике освоить библиотечные функции представленные в `dirent.h`, выполнить практическое задание.

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.

Ход работы.

1. Подключены нужный для работы программы заголовочные файлы

```
#include <stdio.h>
#include <dirent.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
```

2. Создана функция для поиска нужных файлов в директории, полученной из функции main.

```
int letter_file (char* start, char c)
{
    DIR *dir = opendir(start);
    if (dir == NULL)
        return -1;
    struct dirent *entry;
    char current_path[1000];
    int flag = 0;
    strcpy(current_path, start);
    while ((entry = readdir(dir)) && !flag)
    {
        if (!strcmp(entry->d_name, ".") || !strcmp(entry->d_name, ".."))
            continue;
        strcat(current_path, "/");
        strcat(current_path, entry->d_name);
        if (entry->d_type == DT_REG)
        {
            if ((c == (entry->d_name)[0]) && (strlen(entry->d_name) == 5))
            {
                int i;
                for (i = 0; strlen(dir_word[i])>0; i++)
                    if (!strcmp(current_path, dir_word[i]))
                        break;
                if (!strlen(dir_word[i]))
                {
                    strcpy(dir_word[i], current_path);
                    flag = 1;
                }
            }
        }
        else if (entry->d_type == DT_DIR)
        {
            flag = letter_file(current_path, c);
            current_path[strlen(current_path) - strlen(entry->d_name)-1] = '\0';
        }
    }
    closedir(dir);
    return flag;
}
```

3. Создан массив для хранения, наименований файлов, отвечающих условию задачи.

```
char dir_word[100][100];
```

4. Создана функция main, принимающая в качестве аргументов исходную директорию, вызывающая функцию letter_file, и обрабатывающая возвращаемое ей значение.

```
int main (int c, char** vr)
{
    char word[100];
    int err;
    if (c != 2)
    {
        printf ("Wrong input!\n");
        exit(1);
    }
    if (vr[1][strlen(vr[1])-1] == '/')
        vr[1][strlen(vr[1])-1] = '\0';
    scanf ("%s", word);
    for (int i = 0; i < strlen(word); i++)
    {
        err = letter_file(vr[1], word[i]);
        if (err < 0)
        {
            printf ("Wrong directory!\n");
            exit(-1);
        }
        else if (!err)
        {
            printf ("Sorry, but i can do nothing with this word.\n");
            exit (0);
        }
    }
    for (int i = 0; strlen(dir_word[i]); i++)
        printf ("%s\n", dir_word[i]);
    return 0;
}
```

5. Проверка программы на корректность работы

- 1) **Input:** `./hello_world_test`
`HeLLO`
- Output:** `./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`
- 2) **Input:** `./test`
`IamACoolProgrammer`
- Output:** `./test/vowels/set_4/capital/I.txt`
`./test/vowels/set_4/capital/A.txt`
`./test/consonant/set_4/small/m.txt`
`./test/vowels/set_2/capital/A.txt`
`./test/consonant/set_4/capital/C.txt`
`./test/vowels/set_4/small/o.txt`
`./test/vowels/set_2/small/o.txt`
`./test/consonant/set_4/small/l.txt`
`./test/consonant/set_4/capital/P.txt`
`./test/consonant/set_4/small/r.txt`
`./test/vowels/set_5/small/o.txt`
`./test/consonant/set_4/small/g.txt`
`./test/consonant/set_2/small/r.txt`
`./test/vowels/set_4/small/a.txt`
`./test/consonant/set_2/small/m.txt`
`./test/consonant/set_5/small/m.txt`
`./test/vowels/set_4/small/e.txt`
`./test/consonant/set_5/small/r.txt`

Вывод: *В ходе работы был выполнен рекурсивный обход дерева файлов и папок, а так же поиск элементов по определенным критериям.*