

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: РАБОТА С ФАЙЛОВОЙ СИСТЕМОЙ

Студент гр.0382

Диденко Д.В.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2021

Цель работы.

Изучение функций для работы с файловой системой.

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.

!Регистрозависимость

!Могут встречаться файлы, в имени которых есть несколько букв и эти файлы использовать нельзя.

!Одна буква может встречаться один раз.

Основные теоретические положения.

Для работы с файловой системой используется библиотека `dirent.h`. Ниже представлены ее функции.

```
DIR *opendir(char *dirname)
struct dirent *readdir(DIR *ptr)
void rewinddir(DIR *ptr)
int closedir(DIR *ptr)
```

Функция `opendir()` открывает поток каталога и возвращает указатель на структуру типа `DIR`, которая содержит информацию о каталоге. Не следует модифицировать содержимое данной структуры. Функция `closedir()` закрывает поток каталога, на который указывает `ptr`.

Функция `readdir()` возвращает название следующего файла в каталоге. Иными словами, функция `readdir()` читает оглавление каталога по одному файлу за раз. Параметр `ptr` должен указывать на поток каталога, открытый с

помощью opendir(). Структура dirent определена для DOS следующим образом:

```
struct dirent{  
    char d_name[13];  
};
```

Таким образом, после вызова функции readdir() параметр d_name содержит имя следующего файла в каталоге. Для Windows длина d_name равна 260 байтам. Для OS/2 длина равна 256 байтам.

Функция rewinddir() вызывает возвращение в начало каталога, на который указывает ptr и который был предварительно получен с помощью opendir(). Это означает возвращение к первой позиции в каталоге. Благодаря этому каталог может быть прочитан снова.

Функция closedir() в случае успеха возвращает 0 и —1 в противном случае. При неудаче переменная errno устанавливается равной EBADF (недействительный каталог). Функция opendir() возвращает NULL в том случае, если каталог не может быть открыт. При этом переменная errno устанавливается равной либо ENOENT (каталог не найден), либо ENOMEM (недостаточно памяти). Функция readdir() возвращает NULL, когда достигается конец каталога.

Выполнение работы.

Исходный код решения задачи см.в приложении А.

В функции main в переменную tmp записывается путь к каталогу, который предстоит пробежать. Через функцию fopen получается указатель на файл file, в который будет записан результат. В переменную word вводится через fgets список символов, которые совпадают с названиями файлов в указанной директории,которые нужно вывести. Далее в цикле for каждый символ поочередно отправляется в функцию list_dir, которая рекурсивно открывает все вложенные директории и ищет файлы с помощью библиотеки dirent.h. После цикла закрывается file и очищается память, выделенная под word.

Функция `list_dir` принимает на вход путь к директории, в которой производится поиск, символ, указатель на файл, в который производится запись. Описание работы `list_dir`:

Через функцию `opendir` "открываем" директорию, если это удастся, то с помощью `readdir` получается очередной элемент открытой директории. Если удастся, к пути добавляется «/», путь копируется в переменную `dirPath_add` и к этой переменной склеивается название текущего элемента директории. Далее следует проверка, если текущий элемент — нужный файл, его путь записывается в `file`, если нет — рекурсивно вызывается `list_dir` с новым путем `dirPath_add`. Последним действием в цикле получаем следующий элемент директории и все действия повторяются с ним. После цикла закрывается открытая директория.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Hello	/home/danil/Documents/ CProject/hello_world_test/ jdgfdj/kjhgkh/H.txt /home/danil/Documents/CP roject/hello_world_test/qwe rty/e.txt /home/danil/Documents/CP roject/hello_world_test/jdgm dj/l.txt /home/danil/Documents/CP roject/hello_world_test/jdgm dj/l.txt /home/danil/Documents/CP roject/hello_world_test/jdgm dj/O.txt	Программа работает верно
2.	HeO	/home/danil/Documents/	Программа работает

		CProject/hello_world_test/ jdgfdj/kjhgkh/H.txt /home/danil/Documents/CP roject/hello_world_test/qwe rty/e.txt /home/danil/Documents/CP roject/hello_world_test/jdgf dj/O.txt	верно
--	--	---	-------

Выводы.

Были изучены функции для работы с файловой системой.

Разработана программа, которая пробегает по каталогу файлов с помощью функций библиотеки `dirent.h`, находит определенные файлы и записывает пути к ним в файл через функцию `fprintf`.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

void list_dir(char *dirPath, char* word, FILE* file) {
    DIR *dir = opendir(dirPath); // "открываем" директорию
    if (dir){ // если это удалось успешно
        struct dirent *de = readdir(dir); // получаем очередной элемент
открытой директории
        while (de) { // если это удалось
            if(de->d_name[0] != '.'){
                if(dirPath[strlen(dirPath)-1] != '/'){
                    dirPath[strlen(dirPath)+1] = '\\';
                    dirPath[strlen(dirPath)] = '/';
                }
                char dirPath_add[300];
                strcpy(dirPath_add, dirPath);
                strcat(dirPath_add, de->d_name);
                if(word[0] == de->d_name[0] && de->d_name[1] == '.'){
                    fprintf(file, "%s%s\\n", dirPath, de->d_name);
                } else{
                    list_dir(dirPath_add, word, file);
                }
            }
            de = readdir(dir); // снова получаем очередной элемент
открытой директории
        }
    }
    closedir(dir); // не забываем "закрыть" директорию
}

int main(){
    char tmp[100] = "/home/danil/Documents/CProject/hello_world_test
```

```

";
    FILE *file;
    file = fopen("/home/danil/Documents/CProject/result.txt
", "w");
    char* word = (char*)malloc(30*sizeof(char));
    fgets(word, 30, stdin);
    int len = (int)strlen(word);
    for(int i = 0; i < len; i++){
        list_dir(tmp, word+i, file);
    }
    fclose(file);
    free(word);
    return 0;
}

```