

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: ОБХОД ФАЙЛОВОЙ СИСТЕМЫ

Студентка гр. 0382

Чегодаева Е.А.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2021

Цель работы.

Изучение файловых систем на языке Си.

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.

Файл:

➤ Функция `fopen()` открывает файл и возвращает связанный с ним указатель.

→ `FILE *fopen(const char *fname, const char *mode);`

Тип операций, разрешенных над файлом, определяется аргументом `mode`.

Разрешенные для `mode` значения:

- ✓ “r” → Создает файл для чтения (по умолчанию файл открывается как текстовый).
- ✓ “w” → Создает файл для записи (по умолчанию файл открывается как текстовый).
- ✓ “a” → Дописывает информацию к концу файла (по умолчанию файл открывается как текстовый).

- Функция `fclose()` используется для закрытия потока, ранее открытого с помощью `fopen()`. Она сохраняет в файл данные, находящиеся в дисковом буфере, и выполняет операцию системного уровня по закрытию файла. → `int fclose(FILE *fp);`
- Функция записи в файл → `fprint();`

Директория:

Основные функции для работы с деревом файловой системы, объявления которых находятся в заголовочном файле `dirent.h` (также, может понадобиться включить заголовочный файл `sys/types.h`):

- Для того, чтобы получить доступ к содержимому некоторой директории можно использовать функцию → `DIR *opendir(const char *dirname);`

Которая возвращает указатель на объект типа `DIR` с помощью которого можно из программы работать с заданной директорией.

- Тип `DIR` представляет собой поток содержимого директории. Для того, чтобы получить очередной элемент этого потока, используется функция → `struct dirent *readdir(DIR *dirp);`

Она возвращает указатель на объект структуры `dirent`, в котором хранится информация о файле. Основным интерес представляют поля, хранящие имя и тип объекта в директории (это может быть не только "файл" и "папка").

- После завершения работы с содержимым директории, необходимо вызвать функцию → `int closedir(DIR *dirp);`

Передав ей полученный функцией `readdir()` ранее дескриптор.

Выполнение работы.

Функция `main()`:

Объявляется статический массив символом `str` (для хранения пользовательской строки), далее в данный массив записывается полученная строка посредством `fgets()`. Вместе с тем объявляется `way`, в неё записывается строка `"/tmp"` — для передачи в функцию `PrintWay()` данной директории с

целью обработки и получения полного пути. С помощью функции *fopen()* открывается (и создаётся) файл *result.txt* с параметром “w” — возможностью записи в данный файл, возвращает связанный с данным файлом указатель в *result*. Посредством цикла *for()*, где итератор проходится от 0 и до индекса последнего символа строки *str*. Внутри цикла вызывается функция *PrintWay()*. После этого файл *result.txt* закрывается с помощью *fclose()*.

Функция PrintWay():

Принимает на вход один символ, поданной в программу строки (*str*) , указатель на файл с результатом (*res*) и директорию (*path*), являющуюся константой.

Далее посредством функции *DIR *opendir* получаем доступ к содержимому директории, хранящейся в *path* (при первом вызове — *./tmp*). После этого осуществляется проверка на существование директории и наличие прав. Если эти условия выполнены, то объявляется массив *way_p* — для хранения всего пути к файлу, память выделяется динамически посредством *calloc*. Далее получаем следующий элемент потока содержимого открытой директории, используя функцию *struct dirent* de = readdir(dir)*.

Если прошлый шаг выполнен — посредством цикла *while()* в *way_p* копируется наименование директории, которое на данном этапе хранится в *path* с помощью функции *strcpy()*, а также добавляется символ “/” в конце с помощью функции *strcat()*. После этого реализованы две проверки:

- 1) Если данный элемент является файлом (тип совпадает с макросом *DT_REG*),наименование которого совпадает с буквой пользовательской строки, поданной в функцию, и вместе с тем эта буква является полным названием файла, то в файл с результатом записываются обрабатываемая директория и название данного файла через символ “/” с помощью *fprintf()*. И на данном шаге цикл из *main*'а проходит на новую итерацию.

- 2) Если же элемент является директорией (тип совпадает с макросом *DT_DIR*), содержащей файлы, то в *way_p* добавляется название данной (найденной) директории и рекурсивно вызывается функция *PrintWay()* с передачей полученного пути.

Далее получаем новый элемент открытой директории при помощи *readdir()*.

После выхода из цикла обрабатываемая директория закрывается посредством *closedir()*, а также очищается память, выделенная под *way_p*.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Имеется файловая система:

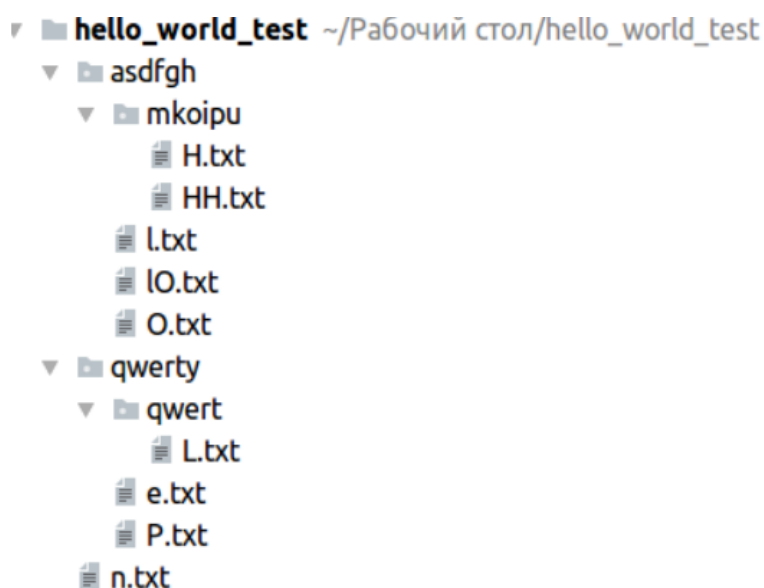


Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	HeLiO	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	Ответ верный.

Выводы.

Были изучены файловые системы на языке Си.

Разработана программа, осуществляющая работу с файлами и директориями. Был реализован поиск файлов по названию, а также запись путей к данным файлам.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb3.c

```
#include <dirent.h>
#include <sys/types.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#define N 100

void PrintWay(char str, FILE *res, const char *path){
    DIR *dir = opendir(path);
    if (!dir){
        return;
    }
    char *way_p=calloc(N, sizeof(char));
    struct dirent* de = readdir(dir);
    while(de){
        strcpy(way_p, path);
        strcat(way_p, "/");
        if ((de->d_type==DT_REG) && (de->d_name[0]==str) && (de->d_name[1]=='.')){
            fprintf(res,"%s/%s\n",path, de->d_name);
        }
        if ((de->d_type==DT_DIR) && (strcmp(de->d_name, ".")!=0) && (strcmp(de->d_name,
"..")!=0)){
            strcat(way_p, de->d_name);
            PrintWay(str, res, way_p);
        }
        de = readdir(dir);
    }
    closedir(dir);
    free(way_p);
}

int main(){
    int i=0;
    char str[N];
    fgets(str, N, stdin);
    FILE *result = fopen("result.txt", "w");
    char* way;
    way = "./tmp";
    for (i=0; i<strlen(str); i++){
        PrintWay(str[i], result, way);
    }
    fclose(result);
    return 0;
}
```