

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Программирование»
Тема: «Обход файловой системы»

Студент гр. 3343

Пивоев Н. М.

Преподаватель

Государкин Я.С.

Санкт-Петербург

2024

Цель работы

Научиться работать с файлами и директориями. Написать, с использованием полученных знаний, программу на языке Си по рекурсивному поиску нужных файлов среди директорий.

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.

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

Сначала считывается строка и открывается файл *result.txt* в режиме редактирования. Для каждого символа строки вызывается функция *findDir* по поиску в директории *tmp* полного пути до нужного файла.

findDir – рекурсивная функция, принимающая в качестве аргументов файл для записи, символ для поиска среди директорий, имя текущей директории (текущий путь). Сперва открывается директория *dir* с помощью *readdir()* и начинается её поэлементное считывание с помощью задания структуры *dirent* и функция *readdir()*. Для каждого элемента идёт две проверки: является ли текущий элемент совпадающим с необходимым и является ли текущий элемент директорией. В первом случае проверяется тип файла и совпадение первого символа с искомым, при прохождении проверок идёт запись полного пути в файл и в поток вывода. Во втором случае идёт проверка на несовпадение с типом *.txt* и со встроенными элементами “.” и “..”, которые являются ссылками на текущую директорию и родительскую, при прохождении проверок выделяется память под путь новой директории, её объём определяется как старый путь + ‘/’ + новый путь + ‘\0’, и идёт повторный вызов функции для полученного пути. После окончания рекурсивных вызовов освобождается память и закрывается директория.

Выводы

Во время выполнения лабораторной работы, мы ознакомились с синтаксисом языка Си по работе с директориями и файлами, а также написали программу по рекурсивной обработке файловой системы и поиску файлов в ней.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

void findDir(FILE* file, char letter, char* dirName) {
    DIR* dir = opendir(dirName);
    if (dir) {
        struct dirent* de = readdir(dir);
        while (de) {
            if (strstr(de->d_name, ".txt") != NULL && de->d_name[0]
== letter && de->d_name[1] == '.') {
                fprintf(file, "%s/%s\n", dirName, de->d_name);
                printf("%s/%s\n", dirName, de->d_name);
            }

            else if (strstr(de->d_name, ".txt") == NULL &&
strcmp(de->d_name, ".") != 0 && strcmp(de->d_name, "..") != 0) {
                char*
                    newDir
=
malloc(strlen(dirName)+strlen(de->d_name)+2);
                sprintf(newDir, "%s/%s", dirName, de->d_name);
                findDir(file, letter, newDir);
                free(newDir);
            }
            de = readdir(dir);
        }
        closedir(dir);
    }
}

int main() {
    char str[1000];
    scanf("%s", str);
    FILE* file = fopen("result.txt", "w");

    for (int i = 0; i < strlen(str); ++i)
        findDir(file, str[i], "./tmp");

    fclose(file);
    return 0;
}
```