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

Студентка гр. 3341 _____ Кузнецова С.Е.

Преподаватель _____ Глазунов С.А.

Санкт-Петербург
2024

Цель работы

Цель работы – ознакомиться с принципом рекурсивного обхода файловой системы и написать программу на языке Си с использованием заголовочного файла dirent.h, который содержит основные функции для работы с деревом файловой системы. Программа будет осуществлять обход файловой системы, анализировать названия всех файлов в нужной директории и записывать полные пути подходящих под условие файлов в файл с результатом.

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

Были подключены заголовочные файлы stdio.h, string.h, а также заголовочный файл dirent.h для работы с файловой системой.

Объявлены функции:

1. int main() – объявляет переменные с путём исследуемой директории и путём файла для записи результата. Принимает на вход слово, на основании которого будут найдены файлы с требуемыми буквами. Открывается файл для записи, циклом перебираются буквы слова, для каждой из букв выполняется функция find_name, находящая файл с именем, равным букве слова. Закрывается файл с результатом.
2. int nec_letter(char* d_name, char letter) – проверяет имя файла на соответствие букве слова.
3. void find_name(FILE* result, char letter, const char *directory) – принимает на вход файл для записи результата, букву и адрес обрабатываемой директории. Функция обходит все файлы в директории. Если перед нами еще одна директория, функция вызывает саму себя. Если файл – его название проверяется на соответствие букве слова с помощью функции nec_letter, при соответствии путь обрабатываемого файла записывается в файл с результатом. Каждая директория впоследствии закрывается.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	HeLlO	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	Тест проверяющей системы e.moevm
2.	HOle	hello_world_test/asdfgh/mkoipu/H.txt hello_world_test/asdfgh/O.txt hello_world_test/asdfgh/l.txt hello_world_test/qwerty/e.txt	Тест на поиск файлов из букв другого слова
3.	LORD	hello_world_test/qwerty/qwert/L.txt hello_world_test/asdfgh/O.txt	Неполное соответствие имён файлов буквам слова, файлы для некоторых букв не найдены, т.к. отсутствуют файлы с такими названиями

Выводы

В ходе выполнения работы были достигнуты поставленные цели: приобретены навыки работы с принципом рекурсивного обхода файловой системы и написана программу на языке Си с использованием заголовочного файла dirent.h, который содержит основные функции для работы с деревом файловой системы. Программа осуществляет обход файловой системы, анализирует названия всех файлов в нужной директории и записывает полные пути подходящих под условие файлов в файл с результатом.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: solution.c

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

#define length 500
#define dot '.'


void find_name(FILE *result, char str, const char *dirPath);

int main() {

	const char *directory = "./tmp";
	const char firstpath[] = "./result.txt";

	int i = 0;
	char word[length];
	scanf("%s", word);

	FILE *result = fopen(firstpath, "w");

	while (word[i]) {
		find_name(result, word[i], directory);
		i++;
	}

	fclose(result);
}

int nec_letter(char* d_name, char letter) {
	return (d_name[0] == letter) && (d_name[1] == dot);
}

void find_name(FILE* result, char letter, const char *directory) {
	if (result) {
		DIR *dir = opendir(directory);
		if (dir) {
			struct dirent *entry = readdir(dir);
			while (entry) {
				char path[length];
				sprintf(path, "%s/%s", directory, entry-
>d_name);

				if (entry->d_type == DT_DIR && entry-
>d_name[0] != dot) {
					find_name(result, letter, path);
				}

				if (entry->d_type == DT_REG &&
nec_letter(entry->d_name, letter)) {
					fprintf(result, "%s\n", path);
				}
```

```
                        entry = readdir(dir);
                    }
                }
                closedir(dir);
            }
        }
```