**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.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 1384 |  | Усачева Д.В. |
| Преподаватель |  | Жангиров Т. Р. |
|  |  |  |

Санкт-Петербург, 2022

Цель работы.

Научиться использовать рекурсию в языке Си.

Задание.

Вариант 3

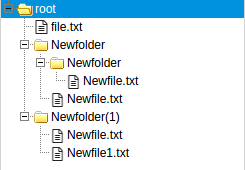
Дана некоторая корневая директория, в которой может находиться некоторое количество папок, в том числе вложенных. В этих папках хранятся некоторые текстовые файлы, имеющие имя вида .txt

В каждом текстовом файле хранится одна строка, начинающаяся с числа вида:

<число><пробел><латинские буквы, цифры, знаки препинания> ("124 string example!")

Требуется написать программу, которая, будучи запущенной в корневой директории, выведет строки из файлов всех поддиректорий в порядке возрастания числа, с которого строки начинаются

**Пример**



root/file.txt: 4 Where am I?  
root/Newfolder/Newfile.txt: 2 Simple text  
root/Newfolder/Newfolder/Newfile.txt: 5 So much files!  
root/Newfolder(1)/Newfile.txt: 3 Wow? Text?  
root/Newfolder(1)/Newfile1.txt: 1 Small text

**Решение:**

1 Small text  
2 Simple text  
3 Wow? Text?  
4 Where am I?  
5 So much files!

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.

Выполнение работы.

Для выполнения поставленного задания было реализовано несколько функций. Было необходимо при помощи рекурсии перебрать все поддиректории и, если встречаем текстовый файл записывать его первую строку в массив. Далее было необходимо отсортировать массив по первому символу (каждая строка начиналась с числа) и записать результат в файл.

Функция read\_dir открывает директорию и рекурсивным способом перебирает все поддиректории и считывает первые строчки из всех текстовых файлов, находящиеся в них. Функция freetxt освобождает занятую массивом динамическую память. Функция writefile перезаписывает файл «result.txt» и записывает строчки массива в этот файл в первоначальном порядке. Функция

comparator создана для сравнения строк.

**ПРИЛОЖЕНИЕ 1**

**ТЕСТИРОВАНИЕ**

Результаты тестирования представлена в таблице 1.

Таблица 1 - Пример тестовых случаев

|  |  |  |
| --- | --- | --- |
| № п/п | Входные данные | Выходные данные |
| 1. | root/file.txt: 4 Where am I? root/Newfolder/Newfile.txt: 2 Simple text root/Newfolder/Newfolder/Newfile.txt: 5 So much files! root/Newfolder(1)/Newfile.txt: 3 Wow? Text? root/Newfolder(1)/Newfile1.txt: 1 Small text | 1 Small text 2 Simple text 3 Wow? Text? 4 Where am I? 5 So much files! |

Вывод.

Были освоены основы работы с рекурсией и обходом файловой системы в языке Си.

**ПРИЛОЖЕНИЕ 2**

**ИСХОДНЫЙ КОД ПРОГРАММЫ**

Название файла: main.c

#include <stdio.h>

#include <dirent.h>

#include <string.h>

#include <stdlib.h>

int n = 0;

void read\_dir(char \*dirname, char \*\*txtarr){

DIR \*dir = opendir(dirname);

struct dirent \*fi;

while(fi = readdir(dir)){

if(fi->d\_type==DT\_DIR && fi->d\_name[0]!='.'){

int senlen = strlen(dirname);

strcat(dirname, "/");

strcat(dirname, fi->d\_name);

read\_dir(dirname, txtarr);

dirname[senlen] = '\0';

}

if(fi->d\_type==DT\_REG){

int senlen = strlen(dirname);

strcat(dirname, "/");

strcat(dirname, fi->d\_name);

FILE \*txt = fopen(dirname, "r");

txtarr[n] = malloc(sizeof(char) \* 50);

fgets(txtarr[n], 50, txt);

strcat(txtarr[n], "\n");

n++;

fclose(txt);

dirname[senlen] = '\0';

}

}

closedir(dir);

}

void freetxt(char \*\*txtarr){

int i;

for(i = 0; i<n; i++){

free(txtarr[i]);

}

free(txtarr);

}

void writefile(char \*\*txtarr){

FILE \*txt = fopen("result.txt", "w");

int i;

for(i = 0; i<n; i++){

fputs(txtarr[i], txt);

}

}

int comparator(const void \*a, const void \*b){

long int aa = atoi(\*((char\*\*) a));

long int bb = atoi(\*((char\*\*) b));

if(aa > bb){

return 1;

}

if(aa < bb){

return -1;

}

else{

return 0;

}

}

int main(){

char dirname[50] = "./root";

char \*\*txtarr = malloc(sizeof(char\*) \* 3000);

read\_dir(dirname, txtarr);

qsort(txtarr, n, sizeof(char\*), comparator);

writefile(txtarr);

freetxt(txtarr);

return 0;

}