

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. 1304

Маркуш А.Е.

Преподаватель

Чайка К. В.

Санкт-Петербург

2022

Цель работы.

Научиться работать с файлами и директориями в языке Си. Изучить и применить рекурсивный обход файловой системы в глубину.

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 Файл с решением должен называться solution.c. Результат работы программы должен быть записан в файл result.txt.

Выполнение работы.

Программа рекурсивно перебирает все файлы в корневой директории, при этом записывает в массив `answArr` содержание файлов. После этого сортирует с помощью компаратора строки в этом массиве и построчно записывает их в файл `result.txt`.

Выводы.

Был изучен принцип работы с файлами, директориями, применен алгоритм рекурсивного обхода файловой системы. Была написана программа, считывающая содержание файлов и сортирующая содержание согласно заданию.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

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

typedef struct{
    char content[1000];
    long int pos;
}file;

int cmp(const void *a, const void *b){
    const file *val_a = a;
    const file *val_b = b;

    int firstnum = val_a->pos;
    int secondnum = val_b->pos;

    if (firstnum > secondnum){
        return 1;
    }
    else if (firstnum < secondnum){
        return -1;
    }
    return 0;
}

void dir_list(const char* dirname, file* answArr){

    DIR* dir = opendir(dirname);

    struct dirent* entity;
    entity = readdir(dir);
    int i;
    file* arr[1000];
    int arriter = 0;

    while (entity != NULL){

        if ((entity->d_type == DT_REG) && (strcmp(entity->d_name, "solution.c") != 0) && (strcmp(entity->d_name, "a.out") != 0)){

            FILE* ptr;
            char ch;
            char path[100] = {0};
```

```

        strcat(path,dirname);
        strcat(path,"/");
        strcat(path,entity->d_name);

        char str[50];
        ptr = fopen(path, "a+");

        if (NULL == ptr) {
            printf("file can't be opened \n");
        }
        while (fgets(str, 50, ptr) != NULL) {
            char num[50];
            char content[100];
            int iternum = 0;
            int itercontent = 0;
            int flag = 1;
            for (i = 0; i < strlen(str); i++) {
                if (isspace(str[i])) {
                    flag = 0;
                    num[iternum] = '\\0';
                }
                if (flag) {
                    num[iternum++] = str[i];
                }
                content[itercontent++] = str[i];
            }
            content[itercontent] = '\\n';
            content[itercontent+1] = '\\0';
            file* cur = calloc(sizeof(file),1);
            strcpy(cur->content,content);
            cur->pos = atoi(num);
            arr[arriter++] = cur;
        }
        fclose(ptr);
    }
    else if ((entity->d_type == DT_DIR) && (strcmp(entity->d_name, ".") != 0) && (strcmp(entity->d_name, "..") != 0)) {
        char path[100] = {0};
        strcat(path,dirname);
        strcat(path,"/");
        strcat(path,entity->d_name);
        dir_list(path,answArr);
    }

    entity = readdir(dir);
}

int answ_size = 0;

```

```

while (answArr[answ_size].pos != 0){
    answ_size++;
}

int k = 0;
for (i = answ_size; i < answ_size+arriter;i++){
    answArr[i] = *arr[k++];
}

closedir(dir);
}

int main(){

    file* answArr = calloc(sizeof(file), 5000);

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

    dir_list("root",answArr);

    int i = 0;
    while(answArr[i].pos!=0){
        i++;
    }

    qsort(answArr, i, sizeof(file),cmp);
    i = 0;
    while(answArr[i].pos!=0){

        fputs(answArr[i].content, fp);
        i++;
    }

    fclose(fp);

    return 0;
}

```