

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

Студент

Докучаев Р.А.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Изучить способы обхода дерева на примере рекурсивного обхода файловой системы UNIX.

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

Основные теоретические положения.

Были использованы заголовочные файлы *stdio.h*, *stdlib.h*, *string.h*, *dirent.h* и *sys/types.h*.

Были использованы функции стандартной библиотеки: *malloc(size_t, size_t)* для выделения места под полные пути до найденных файлов, *FILE* fopen(const char*, const char*)* и *char *fgets(char *str, int num, FILE *stream)* для считывания строки из файла, *qsort* — для быстрой сортировки массива строк. Были использованы *strcpy* и *strcat* из библиотеки *string.h* для работы со строками и редактировании пути к нужным файлам.

Выполнение работы.

1. Были подключены заголовочные файлы *stdio.h*, *stdlib.h*, *string.h*, *dirent.h* и *sys/types.h*

2. Была создана функция *comp* типа *int* с аргументами *const void* a* и *const void* b*, которая служит для сортировки массива *temp*. При помощи *sscanf* считываются числа согласно условию в начале текстового файла до пробела, далее данные числа сравниваются. В зависимости от результата сравнения функция возвращает -1, 0 или 1.
3. Была создана функция *find_dir* типа *void* с аргументами *dir_path* (проверяемая директория), *tmp* (буфер для хранения всех строк из файлов) и *i* (номер строки для строки буфера), которая при помощи рекурсии находит нужные файлы и выносит строки из них в отдельный буфер. После директория закрывается.
4. Была создана функция *int main()*. Были объявлены переменные, пути к директории и буфера строк. Далее происходит поиск файлов и их сортировка функцией быстрой сортировки *qsort*. При помощи функции *fprintf* результат сортировки записывается в файл *result.txt*, после чего файл закрывается при помощи функции *fclose*.

Вывод.

В ходе работы были изучены принципы работы с абстрактной структурой данных при помощи рекурсивного обхода файловой системы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название исходного файла: laba3.c

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

int comp(const void* a, const void* b){
    int num1, num2;
    sscanf(*(char**)a, "%d", &num1);
    sscanf(*(char**)b, "%d", &num2);
    if(num1 > num2) return 1;
    if(num1 == num2) return 0;
    if(num1 < num2) return -1;
}

void find_dir(char* dir_path, char** tmp, int* i){ //В tmp вписываем
    все строки для сортировки, i отвечает за номер элемента массива в tmp
    FILE *f;
    DIR *dir = opendir(dir_path);
    if(dir){
        struct dirent *de;
        while (de = readdir(dir)){
            //Строки для смены адреса директории
            char* address = malloc((strlen(dir_path)+strlen(de-
>d_name)+2)*sizeof(char));
            strcpy(address, dir_path);
            strcat(address, de->d_name);
            if(de->d_type == DT_REG){
                if(f = fopen(address, "r")){
                    tmp[*i] = malloc(32*sizeof(char));
                    if(!fgets(tmp[*i],
32*sizeof(char), f)){
                        continue;
                    }
                    (*i)++;
                    fclose(f);
                }
            }
            else if(de->d_type == DT_DIR){
                if(!strcmp(de->d_name, ".") || !strcmp(de-
>d_name, "..")) //Проверяем, не оказались ли мы в предыдущей директо-
рии
                {
                    free(address);
                    continue;
                }
                char *new_dir_path = mal-
loc((strlen(dir_path)+strlen(de->d_name)+2)*sizeof(char)); //Переходим
в новую директорию, так как не нашли файл для считывание
                strcpy(new_dir_path, dir_path);
```

```

        strcat(new_dir_path, de->d_name);
        strcat(new_dir_path, "/");
        find_dir(new_dir_path, tmp, i);
        free(new_dir_path);
    }
    free(address);
}
closedir(dir);
}
}

int main(){
    char* dir_path = "./root/";
    char* tmp[8192];
    int i = 0;

    find_dir(dir_path, tmp, &i);
    qsort(tmp, i, sizeof(char*), comp);

    FILE *f;
    f = fopen("./result.txt", "w");
    int j;
    for(j = 0; j < i; j++){
        //      fprintf(stderr, "%p\n", tmp[j]);
        fprintf(f, "%s\n", tmp[j]);
    }
    fclose(f);
}

```