**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Программирование»**

ТЕМА: **Строки. Рекурсия, циклы, обход дерева**.

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Эвергрин П.С. |
| Преподаватель |  | Берленко Т. А. |

Санкт-Петербург

2017

**Цель работы:**

Обретение навыков работы с деревом директорий и файлов и считывания содержимого файлов

**Задание:**

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 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.

**Ход работы:**

1. Подключены необходимые библиотеки и объявлены константы

//Подключение библиотек

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/types.h>

#include <dirent.h>

//Определение констант

#define PATH\_LEN 500

#define WRD\_LEN 200

#define WordNum 100

1. Затем была написана рекурсивная функция обхода директорий и файлов.

void ArrPrint(char \*curr\_path, char\*\* str\_word, int \*str\_word\_len)

{

strcat(curr\_path, "/");

DIR \* dir = opendir(curr\_path);

struct dirent \*obj = readdir(dir);

char file\_path[PATH\_LEN];

if(dir)

{

while(obj)

{

if(obj->d\_type == DT\_DIR && strcmp(obj->d\_name, ".") && strcmp(obj->d\_name, ".."))

/\*d\_name содержит начало символьного массива, дающего имя этой структуры каталога.

Это имя завершается нулем и может содержать самое большее MAXNAMLEN символов.\*/

{

int pathLen;

pathLen=strlen(curr\_path);

strcat(curr\_path, obj->d\_name);

ArrPrint(curr\_path, str\_array, str\_array\_len);

curr\_path[pathLen] = '\0'; //"обрубание" части названия пути

}

if(obj->d\_type == DT\_REG && strstr(obj->d\_name, ".txt")) //найден файл формата .txt

{

strcpy(file\_path, curr\_path);

strcat(file\_path, obj->d\_name);

FILE \*f = fopen(file\_path, "r"); //Открываем текстовый файл на чтение

if(f)

{

fgets(str\_array[\*str\_array\_len], WRD\_LEN, f);

(\*str\_word\_len)++;

fclose(f);

}

}

obj = readdir(dir);

}

}

closedir(dir);

}

1. Код основной части

int main()

{

char path[PATH\_LEN] = ".";

int str\_word\_len = 0;

char \*\* str\_word = (char\*\*)malloc(WordNum \* sizeof(char\*));

for(int i = 0; i < WordNum; i++)

str\_word[i] = (char\*)malloc(sizeof(char)\*WRD\_LEN);

ArrPrint(path, word\_array, &str\_word\_len);

for(int i = 0; i < str\_word\_len; i++)

for(int j = 0; j < str\_word\_len -i - 1; j++)

if(atoi(str\_word[j]) > atoi(str\_word[j+1]))

{

char\* str = str\_word[j];

str\_word[j] = str\_word[j + 1];

str\_word[j + 1] = str;

}

//Вывод на экран

for(int i = 0; i < str\_word\_len; i++)

printf("%s", str\_word[i]);

//Освобождение памяти

for(int i = 0; i < WordNum; i++)

free(str\_word[i]);

free(str\_word);

return 0;

}

1. Затем код работы вместе с отчетом были загружены на github:

* Создана новая ветка с помощью команды *git checkout –b evergreen\_sem2\_lab3*
* Добавлены файлы для загрузки с помощью команды *git add*
* Добавлен комментарий коммита командой *git commit –m “Lab3 ”*
* Файлы были загружены командой *git push origin evergreen \_sem2\_lab3*

**Вывод:** В ходе лабораторной работы были получены навыки работы с деревом директорий и файлов, а также навыки считывания данных из текстовых файлов.