**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе 3**

**по дисциплине «Программирование»**

Тема: Работа с текстовыми файлами и поиск в папках на С.

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Запевалов А.И. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

**Цель работы.**

Научиться работать с файловой системой, используя возможности языка С.

**Задание.**

Дана некоторая корневая дирректория, 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 дирректории, выведет строки из файлов всех поддиректорий в порядке возрастания числа, с которого строки начинаются.

**Ход работы.**

1. На сайте Stepik.org выполним задание Лабораторная работа №3 в блоке 12.Строки. Рекурсия, циклы, обход дерева курса Программирование(ЛЭТИ).

2. Рассмотрим написание основной функции:

int poisk(char \*homedir, char\*\* arr, int i)

{

char buf[100];

DIR \*d;

struct dirent \*dir;

d = opendir(homedir);//opens a directory

if( d == NULL )

{

return 0;

}

while((dir=readdir(d))) //reading contents of directory

{

if( strcmp( dir->d\_name, "." ) == 0 ||

strcmp( dir->d\_name, ".." ) == 0 )//checking if home directory is a root or lower than root

{

continue;

}

int path\_len = strlen(homedir);//saves length of current directory path

strcat(homedir,"/");

strcat(homedir,dir->d\_name);//adding a new folders' name to previous path

if (dir->d\_type == DT\_DIR)

{

i=poisk(homedir, arr, i);//if current directory contains other directories, it calls function 'poisk' again

}

else if (strstr(dir->d\_name, ".txt"))//if contained file has a .txt format, opens it

{

FILE \*file;

file = fopen(homedir, "r");

fgets(buf, 90, file);

strcpy(arr[i], buf);//reads file contents to array

i++;

memset(buf, 0, 100);

fclose(file);

}

homedir[path\_len] = '\0';//turns back to previous directory

}

closedir(d);

return i;

}

В этой функции происходит открытие папки и чтение ее содержимого. Если содержимым оказывается папка, то она также рекурсивно открывается этой же функцией. Если файлом формата .txt — то производится открытие, чтение и запись текста из файла в массив. Эти действия производятся, пока функция readdir возвращает указатель на элементы типа struct dirent. Функция возвращает индекс следующего (свободного) элемента заполняемого массива, чтобы при ее рекурсивном вызове не затирались уже занятые строки в массиве.

3. Исходный код программы:

#include <dirent.h>

#include <sys/types.h>

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define N 20

int comp(const void\* a, const void\* b)

{

return atoi(\*(char\*\*)a)-atoi(\*(char\*\*)b);//compares content of strings, casted to int

}

int poisk(char \*homedir, char\*\* arr, int i)

{

char buf[100];

DIR \*d;

struct dirent \*dir;

d = opendir(homedir);//opens a directory

if( d == NULL )

{

return 0;

}

while((dir=readdir(d))) //reading contents of directory

{

if( strcmp( dir->d\_name, "." ) == 0 ||

strcmp( dir->d\_name, ".." ) == 0 )//checking if home directory is a root or lower than root

{

continue;

}

int path\_len = strlen(homedir);//saves length of current directory path

strcat(homedir,"/");

strcat(homedir,dir->d\_name);//adding a new folders' name to previous path

if (dir->d\_type == DT\_DIR)

{

i=poisk(homedir, arr, i);//if current directory contains other directories, it calls function 'poisk' again

}

else if (strstr(dir->d\_name, ".txt"))//if contained file has a .txt format, opens it

{

FILE \*file;

file = fopen(homedir, "r");

fgets(buf, 90, file);

strcpy(arr[i], buf);//reads file contents to array

i++;

memset(buf, 0, 100);

fclose(file);

}

homedir[path\_len] = '\0';//turns back to previous directory

}

closedir(d);

return i;

}

int main() //creates array of strings, calls function to fill it, sorts strings.

{

char\*\* arr=(char\*\*)malloc(N\*sizeof(char\*));//creates array of pointers to strings

int j=0, i;

for (i=0; i<N; i++)

{

arr[i]=(char\*)malloc(N\*sizeof(char));//creates string for each pointer

}

char homedir[100];

strcpy(homedir, "./root"); //sets home directory

poisk(homedir, arr, 0);//calls function to fill array

qsort(arr, N, sizeof(char\*), comp);//sorting array

while(arr[j])

{

printf("%s", arr[j]);//output

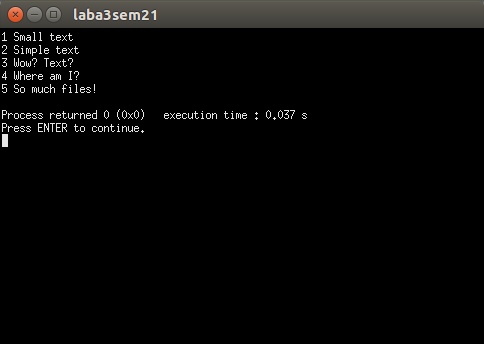
j++;

}

return 0;

}

4. Результат работы программы:

**Вывод.**

Выполнение данной лабораторной работы позволило освоить и закрепить на практике написание программы, выполняющей рекурсивный обход дерева папок и работающей с файлами, содержащимися в этих директориях.