**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 |  | Малышенко Ю.И. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

Цель работы: 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

**Алгоритм программы**

**Открытие директории**

**Создаем структуры для каждого элемента директории**

**Элемент директории**

**Нет**

**Да**

**Файл (.txt)**

**Папка**

**Пропуск**

**+d\_name**

**+"/"**

**+d\_name**

**+"/"**

**Рекурсия для новой директории**

**Функция printFile\_**

**Открытие файла**

**Запись содержимого в список**

**Взять из строки первое число**

**Занести в переменную**

**Закрытие файла**

**\*end**

**Вывод содержимого файлов в порядке возрастания .**

**Функция getIndex обеспечивает порядок вывода.**

**Ход работы.**

1. **Описание стуктур.**

#include <stdio.h>

#include <string.h>

#include <sys/types.h>

#include <dirent.h>

#include <ctype.h>

#include <stdlib.h>

typedef struct MyStruct {

int num;

char c[100];

} MyStruct;

typedef struct MyStructList {

MyStruct str[100];

int size;

} MyStructList;

MyStruct\* getIndex(MyStructList\* list, int i) {

for(int x = 0; x < list->size; x++) {

if(list->str[x].num == i) {

return &(list->str[x]);

}

}

return 0;

}

1. **Функции для работы с поиском подходящей строки.**

MyStruct\* getIndex(MyStructList\* list, int i) {

for(int x = 0; x < list->size; x++) {

if(list->str[x].num == i) {

return &(list->str[x]);

}

}

return 0;

}

1. **Функции для работы с файлом.**

Создается массив для записи содержимого в файлах. Открываем файл для чтения. Записываем содержимое файла в массив. Копируем этот массив в массив list->str[list->size].c. Считывание содержимое массива, разбиваем строчку на токены функцией strtok (по пробелам и \n), далее мы рассматриваем элементы посимвольно функцией isdigit(Возвращает ненулевое значение, если её аргумент является десятичной цифрой, в противном случае возвращается нуль) и преобразуем число с помощью функции atoi и записываем в number. Сохраняет number в list->str[list->size].nim. Увеличиваем size и закрываем файл.

void printFile\_(const char \*path, MyStructList\* list)

{

char arr[100];

int number = 0;

FILE \*f = fopen(path, "r");

if(f)

{

while(fgets(arr, 100, f))

{

strcpy(list->str[list->size].c, arr);

char\* arrPtr = strtok(arr, " ");

if(isdigit(\*arrPtr)) {

number = atoi(arrPtr);

list->str[list->size].num=number;

}

list->size++;

}

fclose(f);

}

}

1. **Функции для работы с директорией.**

Создается массив для записи нашего пути. Копируем в строку текущую начальную директорию. Создаем структуры dirrent для каждого встречающегося элемента директории. Если встречается файл, то мы записываем ее путь и приписываем «/» и имя файла. Вызывается функция для считывания файла. После прочтения фала мы рассматриваем дальше нашу директорию. Если встречается папка, то мы записываем ее путь и приписываем «/» и имя файла. Вызываем функцию с увеличением глубины(рекуррентная функция)

void dirTraveler(const char \*startDir, const int lvl, MyStructList\* list)

{

char path[1000];

strcpy(path, startDir);

DIR \*dir=opendir(path);

if(dir)

{

struct dirent \*de = readdir(dir);

while(de)

{

if(de->d\_type == DT\_REG && strstr(de->d\_name,".txt"))

{

int path\_len = strlen(path);

strcat(path, "/");

strcat(path, de->d\_name);

printFile\_(path, list);

path[path\_len] = '\0';

}

if(de->d\_type == DT\_DIR && 0!=strcmp(".", de->d\_name) && 0!=strcmp("..", de->d\_name))

{

int path\_len = strlen(path);

strcat(path, "/");

strcat(path, de->d\_name);

dirTraveler(path, lvl+1, list);

path[path\_len] = '\0';

}

de = readdir(dir);

}

}

closedir(dir);

}

**3.Функция main()**

В main происходит вызов функции dirTraveler «.» - (текущая директория, глубина, список) и вывод содержание файлов в порядке возрастания.

int main()

{

MyStructList\* list = (MyStructList\*) malloc(sizeof(MyStructList));

list->size = 0;

dirTraveler(".", 0, list);

MyStruct\* temp = 0;

for(int i = 1; i < 100; i++) {

if((temp = getIndex(list, i)) != 0)

printf("%s\n", temp->c);

}

return 0;

}

1. **Исходный код.**

#include <stdio.h>

#include <string.h>

#include <sys/types.h>

#include <dirent.h>

#include <ctype.h>

#include <stdlib.h>

typedef struct MyStruct {

int num;

char c[100];

} MyStruct;

typedef struct MyStructList {

MyStruct str[100];

int size;

} MyStructList;

MyStruct\* getIndex(MyStructList\* list, int i) {

for(int x = 0; x < list->size; x++) {

if(list->str[x].num == i) {

return &(list->str[x]);

}

}

return 0;

}

void printFile\_(const char \*path, MyStructList\* list)

{

char arr[100];

int number = 0;

FILE \*f = fopen(path, "r");

if(f)

{

while(fgets(arr, 100, f))

{

strcpy(list->str[list->size].c, arr);

char\* arrPtr = strtok(arr, " ");

if(isdigit(\*arrPtr)) {

number = atoi(arrPtr);

list->str[list->size].num=number;

}

list->size++;

}

fclose(f);

}

}

void dirTraveler(const char \*startDir, const int lvl, MyStructList\* list)

{

char path[1000];

strcpy(path, startDir);

DIR \*dir=opendir(path);

if(dir)

{

struct dirent \*de = readdir(dir);

while(de)

{

if(de->d\_type == DT\_REG && strstr(de->d\_name,".txt"))

{

int path\_len = strlen(path);

strcat(path, "/");

strcat(path, de->d\_name);

printFile\_(path, list);

path[path\_len] = '\0';

}

if(de->d\_type == DT\_DIR && 0!=strcmp(".", de->d\_name) && 0!=strcmp("..", de->d\_name))

{

int path\_len = strlen(path);

strcat(path, "/");

strcat(path, de->d\_name);

dirTraveler(path, lvl+1, list);

path[path\_len] = '\0';

}

de = readdir(dir);

}

}

closedir(dir);

}

int main()

{

MyStructList\* list = (MyStructList\*) malloc(sizeof(MyStructList));

list->size = 0;

dirTraveler(".", 0, list);

MyStruct\* temp = 0;

for(int i = 1; i < 100; i++) {

if((temp = getIndex(list, i)) != 0)

printf("%s\n", temp->c);

}

return 0;

}

**Вывод**:

В ходе лабораторной работы изучены методы работы с директориями и файлами. Написана программа, которая выведет строки из файлов всех поддиректорий в порядке возрастания числа, с которого строки начинаются