**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 дерева

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Жахин А.А. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

**Цель работы:** Требуется написать программу, выводящую последовательность полных путей файлов, имена которых образуют строку, введенную пользователем. На вход программе подается строка. 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 качестве имени файла используется символ латинского алфавита.

**Ход работы:**

В функции *main* создается массив *arr* в который записываются символы из входного потока. Используя цикл *for*, функция dirTraveler принимает на вход текущую директорию “.”, символ из массива *arr,* и массив *chek*. Если возвращаемое значение функции *dirTraveler* ноль, то программа выводит сообщение “error” и завершает свою работу.

|  |
| --- |
| int main()  {  char arr[11];  char check[1000]="";  fgets(arr, 10, stdin);  for (int i = 0; i < (strlen(arr)-1); i++)  {  if(dirTraveler(".", arr[i], check)==0)  {  printf("error\n");  return 0;  }  }  return 0;  } |

В функции *dirTraveler* создается массив *path,* в который записываются пути до файлов.

Текущая директория записывается в массив *path.* Далее открывается текущая директория.

|  |
| --- |
| int dirTraveler(const char \*startDir, char symbol, char \*check)  {  char path[1000];  strcpy(path, startDir);  DIR \*dir=opendir(path); |

Если открытие директории прошло успешно, то для каждого элемента директории создается структура *dirent,* которая заполняется информацией для соответсвующего элемента директории. Далее используется цикл *while*. Если тип найденного объекта является файлом и в названии файла присутствует расширение *.txt* и первый символ найденного файла совпадает с символом искомого файла, то выполняются следующие действия: создается переменная *path\_len* типа *int* в которую записывается длина массива *path.* Затем в массив path дозаписывается символ “/” и название файла. Затем проверяется использовался ли этот файл ранее. Если не использовался, то массив *path* дозаписывается в массив *check*. Далее выводится путь до данного файла и в конец массива *path* добавляется ‘\0’. Далее директория закрывается и функция возвращает 1.

|  |
| --- |
| if(dir)  {  struct dirent \*de = readdir(dir);  while(de)  {  if(de->d\_type == DT\_REG && strstr(de->d\_name,".txt") &&  de->d\_name[0] == symbol)  {  int path\_len = strlen(path);  strcat(path, "/");  strcat(path, de->d\_name);  if(strstr(check, path))  {  return 0;  }  else  {  strcat(check, path);  printf("%s\n", path);  path[path\_len] = '\0';  closedir(dir);  return 1;  }  } |

Если найденный файл является директорией, то необходимо условие, чтобы поиск нужного файла не выполнялся в предыдущих директориях. Далее создается переменная path\_*len* куда записывается длина массива path и в массив path дозаписывается “/” и название текущей директории. Далее рекурсивно вызывается функция *dirTraveler*.

Если открытие первоначальной директории не произошло, то выводится сообщение “*error*” и функция возвращает 0. Затем закрывается текущая директория и функция возвращает 0 и завершает свою работу.

|  |
| --- |
| 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);  if(dirTraveler(path, symbol, check)==1)  return 1;  path[path\_len] = '\0';  }  de = readdir(dir);  }  }  else  {  printf("error");  return 0;  }  closedir(dir);  return 0;  } |

**Исходный код:**

|  |
| --- |
| #include <stdio.h>  #include <string.h>  #include <sys/types.h>  #include <dirent.h>  #include <stdbool.h>  int dirTraveler(const char \*startDir, char symbol, char \*check)  {  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") && de->d\_name[0] == symbol)  {  int path\_len = strlen(path);  strcat(path, "/");  strcat(path, de->d\_name);  if(strstr(check, path))  {  return 0;  }  else  {  strcat(check, path);  printf("%s\n", path);  path[path\_len] = '\0';  closedir(dir);  return 1;  }  }  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);  if(dirTraveler(path, symbol, check)==1)  return 1;  path[path\_len] = '\0';  }  de = readdir(dir);  }  }  else  {  printf("error");  return 0;  }  closedir(dir);  return 0;  }  int main()  {  char arr[11];  char check[1000]="";  fgets(arr, 10, stdin);  for (int i = 0; i < (strlen(arr)-1); i++)  {  if(dirTraveler(".", arr[i], check)==0)  {  printf("error\n");  return 0;  }  }  return 0;  } |

**Вывод:** в ходе лабораторной работы была написана программа, выводящая последовательность полных путей файлов, имена которых образуют строку, введенную пользователем.