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.

Студентка гр. 0382 _____ Михайлова О.Д.

Преподаватель _____ Берленко Т.А.

Санкт-Петербург
2021

Цель работы.

Изучение основных функций для работы с деревом файловой системой.

Задание.

Вариант 4.

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.

На вход программе подается строка. Требуется найти и вывести последовательность полных путей файлов, имена которых образуют эту строку.

! Регистрозависимость

! Могут встречаться файлы, в имени которых есть несколько букв и эти файлы использовать нельзя.

! Одна буква может встречаться один раз.

Основные теоретические положения.

- FILE * fopen (const char * fname, const char * modeopen) – функция открывает файл, имя которого указано в параметре fname и связывает его с потоком, который может быть идентифицирован для выполнения различных операций с файлом. Операции с потоком, выполнение которых разрешено определяются параметром modeopen. Некоторые возможные операции:

✶ «r» - режим открытия файла для чтения. Файл должен существовать;

✶ «w» - режим создания пустого файла для записи. Если файл с таким именем уже существует его содержимое стирается, и файл рассматривается как новый пустой файл;

✶ «a» - дописать в файл. Операция добавления данных в конец файла. Файл создается, если он не существует.

Функции для работы с деревом файловой системы, объявления которых находятся в заголовочном файле dirent.h:

- DIR *opendir(const char *dirname) – функция, использующаяся для того, чтобы получить доступ к содержимому некоторой директории, возвращает указатель на объект типа DIR с помощью которого можно из программы работать с заданной директорией. Тип DIR представляет собой поток содержимого директории.
- struct dirent *readdir(DIR *dirp) – функция, использующаяся для того, чтобы получить очередной элемент этого потока, возвращает указатель на объект структуры dirent, в котором хранится информация о файле.
- int closedir(DIR *dirp)- функция, которую необходимо вызвать после завершения работы с содержимым директории. Ей передается полученный функцией readdir() ранее дескриптор.

Выполнение работы.

Переменные:

- char* name – символьный массив, в который записывается путь директории;
- char str[200] – символьный массив размера 200;
- FILE *result – переменная, в которую записывается файловый дескриптор;
- char *next – символьный массив, в который записывается путь к файлу;
- DIR *dir – переменная, в которую записывается информация о директории;
- struct dirent* de – переменная, в которую записывается указатель на структуру, возвращаемый функцией readdir;

- int len – переменная, в которую записывается текущая длина строки next.

Функции:

Функция **file_path(char *path, int i, char *str, FILE *file)** принимает на вход рассматриваемую директорию path, строку str, введенную пользователем, индекс i элемента строки str и файл file, в который записывается результат. Сначала для строки next динамически выделяется память с помощью функции malloc, для использования которой подключена стандартная библиотека stdlib.h. Далее с помощью функций strcpy и strcat в строку next копируются содержимое строки path и добавляется «/». Для работы с этими функциями подключена стандартная библиотека string.h. После этого происходит открытие директории path и чтение одного ее элемента. С помощью цикла while обрабатывается каждый элемент данной директории. Если он является файлом и его название совпадает с элементом str[i], то его путь записывается в файл result.txt. Если очередной элемент является директорией, то сначала к строке next добавляется его имя, а затем для него вызывается функция file_path. В конце вызывается функция closedir, которая закрывает директорию.

В функции **main()** с помощью функции fgets считывается строка в переменную str. Далее для записи создается пустой файл result.txt. С помощью цикла for для каждого элемента строки, введенной пользователем, вызывается функция file_path, записывающая в result.txt пути файлов, названия которых образуют эту строку. После этого с помощью функции fclose файл закрывается.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1 для следующей директории:

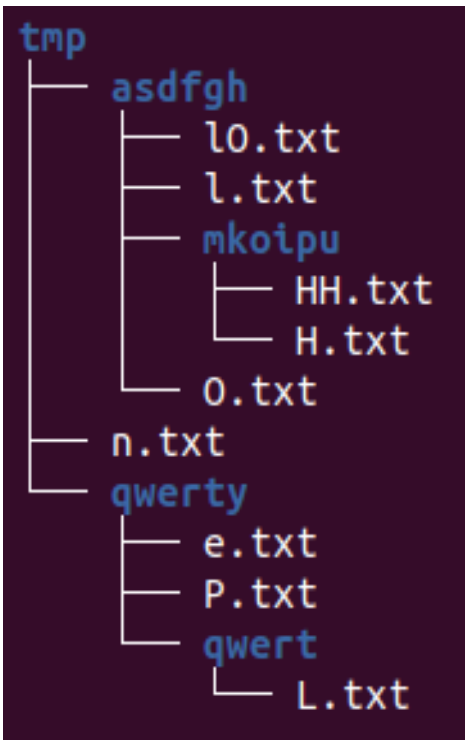

Рисунок 1 - Директория, используемая для тестирования

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	HeLlO	./tmp/asdfgh/mkoipu/H.txt ./tmp/qwerty/e.txt ./tmp/qwerty/qwert/L.txt ./tmp/asdfgh/l.txt ./tmp/asdfgh/O.txt	Результат верный

Выводы.

Были изучены основные функции для работы с файловой системы.

Разработана программа, которая считывает строку, обрабатывает директорию и выводит пути файлов, названия которых образуют эту строку. В результате работы программы создается файл, в котором записан этот вывод.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: pr_lb_3.c

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

void file_path(char *path, int i, char *str, FILE *file){
      char *next = malloc(200*sizeof(char));
      strcpy(next, path);
      strcat(next, "/");
      DIR *dir = opendir(path);
      if (!dir){
            return;
      }
      struct dirent* de = readdir(dir);
      while(de){
            if (de->d_type == DT_REG){
                  if (str[i] == de->d_name[0] && de->d_name[1] == '.'){
                        fprintf(file, "%s/%s\n", path, de->d_name);
                  }
            }
            if (de->d_type == DT_DIR && strcmp(de->d_name, ".")!=0 &&
strcmp(de->d_name, "..")!=0){
                  int len = strlen(next);
                  strcat(next, de->d_name);
                  file_path(next, i, str, file);
                  next[len] = '\0';
            }
            de = readdir(dir);
      }
      closedir(dir);

}

int main(){
      char* name = "./tmp";
      char str[200];
      fgets(str, 200, stdin);
      FILE *result = fopen("result.txt", "w");
      int i;
      for (i=0; i<strlen(str); i++){
            file_path(name, i, str, result);
      }
      fclose(result);
      return 0;
}
```