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.

Студент гр. 1304 _____ Стародубов М.В.

Преподаватель _____ Чайка К.В.

Санкт-Петербург
2022

Цель работы.

Научиться взаимодействовать с файловой системой с помощью языка программирования Си. Научиться обходить дерево файловой системы с помощью рекурсии.

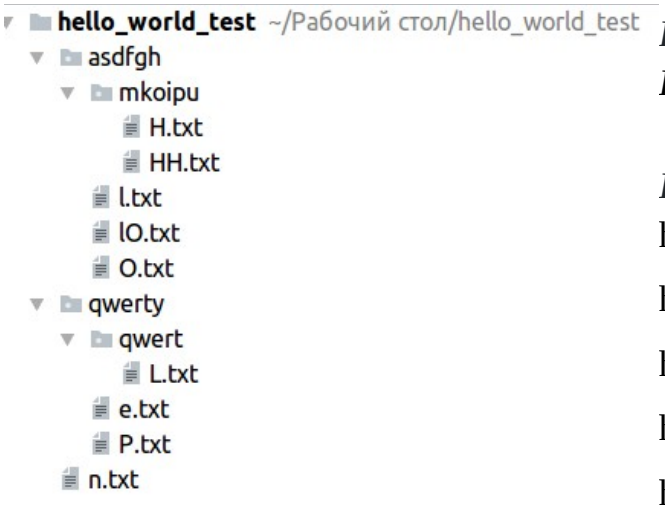
Задание.

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 .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.

Пример



Входная строка:
HeLlO

Правильный ответ:
hello_world_test/asdfgh/mkoipu/H.txt
hello_world_test/qwerty/e.txt
hello_world_test/qwerty/qwert/L.txt
hello_world_test/asdfgh/l.txt
hello_world_test/asdfgh/O.txt

! Регистрозависимость
! Могут встречаться файлы, в имени которых есть несколько букв и эти файлы использовать нельзя.
! Одна буква может встречаться один раз.

Выполнение работы.

Для хранения пути к директории была создана структура *Buffer*, содержащая в себе указатель на выделенную под строку память *content*, а также резмер выделенной памяти *size*. Для взаимодействия с динамической памятью в данной структуре была создана функция *extendBuffer*. Данная функция принимает на вход указатель на структуру *buffer* и выполняет расширение памяти, хранящейся по указателю в поле *content*. При успешном расширении памяти функция возвращает знаяение 0, иначе 1.

Функция *main* начинает выполнение с создания переменной *required_path_name* типа *struct Buffer* и определния знаяений ее полей. Данная переменная булет использоваться для записи пути к найденному файлу. Далее происходит объявление строки *file_name*, в нее будет записано название файла вида «_.txt», где на месте символа «_» будет стоять один из символов, полученых на вход. Далее происходит открытие файла *result.txt*, в который будет производиться запись ответа. Далее в цикле первый символ строки *file_name* последовательно заменяется на вводимые символы и с помощью функции *searchingFolders* происходит поиск файла с соответствующим именем начиная с директории *./tmp*. Если файл был найден, то первый символ строки *file_name* заменяется на следующий и цикл продолжается, иначе программа завершается.

Функция *searchingFolders* принимает на вход указатель на структуру *struct Buffer directory_name_buffer*, в которой храниться название директории, которую необходимо открыть, и строку *file_name*, в которой храниться имя искомого файла. После нахождения файла, путь к нему будет сохранен в структуре. Функция начинает выполнение с открытия необходимой директории, далее, используя переменную current_file, в цикле происходит перебор всех файлов в директории. Если файл является регулярным файлом и его имя

совпадает с именем искомого файла, то к имени директории дописывается имя найденного файла, происходит закрытие директории и функция возвращает 1. Если файл является папкой и его имя не совпадает с именами «.» и «..», то название данной папки дописывается к названию данной директории и происходит рекурсивный поиск внутри данной папки с помощью функции *searchingFolders*, если функция вернула 1, то происходит закрытие теркщей директории и функция возвращает 1. Иначе название директории становиться прежним (каким оно было до дописывания имени найденной папки) и цикл продолжается. Если цикл завершится, то произойдет закрытие текущей директории и функция вернет 0. Если в процессе выполнения функции произойдет ошибка (не выделиться пдополнительная память под имя директории или директория не откроется), то функция вернет 0.

Выводы.

В ходе выполнения данной лабораторной раобты была изучена работа с файловой системой с использованием языка программирования Си. Была создана прогримма, осуществляющая обход дерева файловой системы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <stdlib.h>
#include <stdio.h>
#include <dirent.h>
#include <string.h>

#define BASE_SIZE 300

struct Buffer
{
	char *content;
	int size;
};

int extendBuffer(struct Buffer *buffer)
{
	buffer->size += BASE_SIZE;
	char *tmp = realloc(buffer->content, buffer->size);
	if (tmp)
	{
		buffer->content = tmp;
		return 0;
	}
	return 1;
}

int searchingFolders(struct Buffer *directory_name_buffer, char *file_name)
{
	char *current_directory_name = directory_name_buffer->content;
	DIR *current_directory = opendir(current_directory_name);
	if (!current_directory) return 0;

	struct dirent *current_file;

	while (current_file = readdir(current_directory))
	{
		if (current_file->d_type == DT_REG && !strcmp(current_file-
>d_name, file_name))
		{
			if (strlen(current_directory_name)
+strlen(current_file->d_name)+2 > directory_name_buffer->size &&
extendBuffer(directory_name_buffer))
			{
				closedir(current_directory);
				return 0;
			}

			strcat(current_directory_name, "/");
			strcat(current_directory_name, current_file-
>d_name);
```

```
                        closedir(current_directory);
                        return 1;
                }

                if (current_file->d_type == DT_DIR && strcmp(current_file-
>d_name, ".") && strcmp(current_file->d_name, ".."))
                {
                        if (strlen(current_directory_name)
+strlen(current_file->d_name)+2 > directory_name_buffer->size &&
extendBuffer(directory_name_buffer))
                        {
                                closedir(current_directory);
                                return 0;
                        }

                        int dir_name_length =
strlen(current_directory_name);

                        strcat(current_directory_name, "/");
                        strcat(current_directory_name, current_file-
>d_name);

                        int out = searchingFolders(directory_name_buffer,
file_name);

                        if (out)
                        {
                                closedir(current_directory);
                                return 1;
                        }

                        current_directory_name[dir_name_length] = '\0';
                }
        }

        closedir(current_directory);
        return 0;
}

int main()
{
        struct Buffer required_path_name;
        required_path_name.content = malloc(BASE_SIZE);
        if (!required_path_name.content) return 0;
        required_path_name.size = BASE_SIZE;

        char file_name[strlen("_.txt")+1];
        strcpy(file_name, "_.txt");

        FILE *result_file = fopen("./result.txt", "w");

        scanf("%c", &file_name[0]);
        while (file_name[0] != '\n')
        {
                strcpy(required_path_name.content, "./tmp");
                if (searchingFolders(&required_path_name, file_name))
                {
```

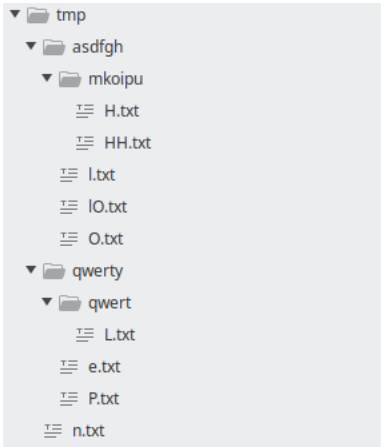
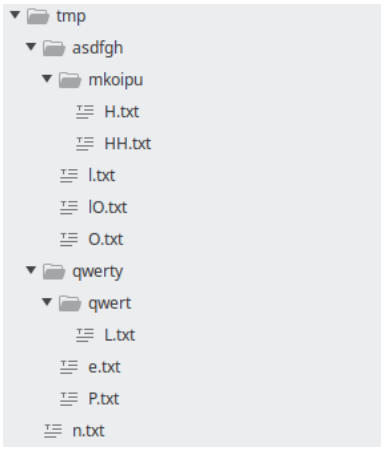
```
                            fprintf(result_file, "%s\n",
required_path_name.content);
                            scanf("%c", &file_name[0]);
                    } else
                    {
                            free(required_path_name.content);
                            fclose(result_file);
                            return 0;
                    }
          }

          free(required_path_name.content);
          fclose(result_file);
          return 0;
}
```

ПРИЛОЖЕНИЕ Б

ТЕСТИРОВАНИЕ

Данные тестирования представлены в таблице 1.

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 корректен
2.	 HOPe	./tmp/asdfgh/mkoipu/H.txt ./tmp/asdfgh/O.txt ./tmp/qwerty/P.txt ./tmp/qwerty/e.txt	Результат корректен

3.	 RecUrsiOn	./tmp/qwerty/qwert/lnop/R.txt ./tmp/qwerty/e.txt ./tmp/asdfgh/mkoipu/c.txt ./tmp/asdfgh/kroo/U.txt ./tmp/asdfgh/mkoipu/tond/r.txt ./tmp/asdfgh/kroo/s.txt ./tmp/qwerty/qwert/lnop/i.txt ./tmp/asdfgh/O.txt ./tmp/n.txt	Результат корректен