МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
"ЛЭТИ" ИМ. В.И.УЛЬЯНОВА(ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЁТ

по лабораторной работе №3
по дисциплине «Программирование»
Тема: Обход файловой системы.

Студент гр. 1304 _____ Мусаев А.И.

Преподаватель _____ Чайка К.В.

Санкт-Петербург
2022

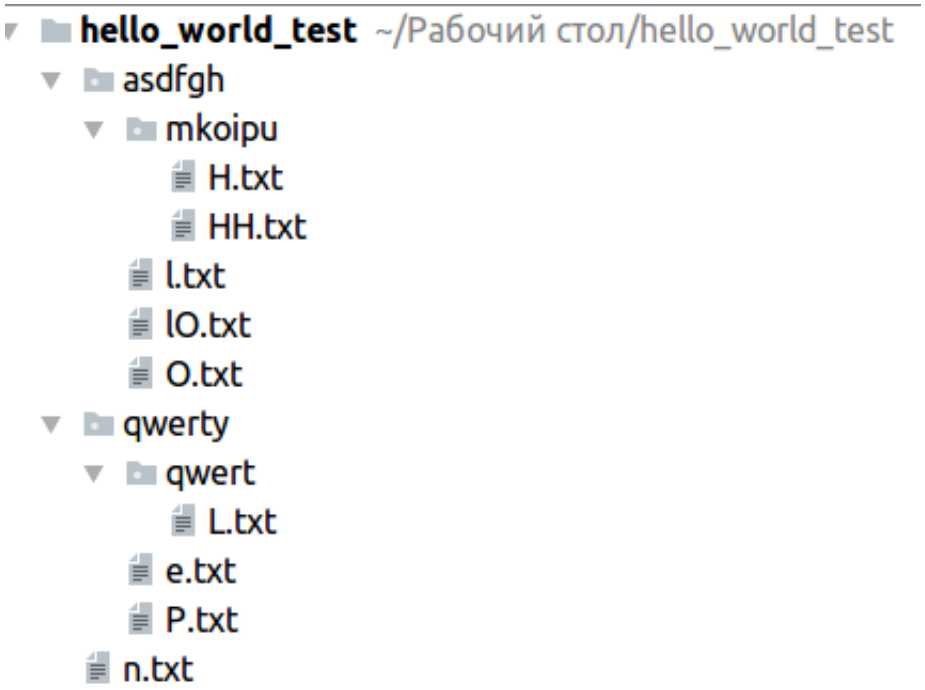
Цель работы:

Целью данной лабораторной работы является изучение работы с файловой системой на язык C и изучение рекурсивного обхода файловой системы.

Задание:

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

Ваше решение должно находиться в директории ***/home/box****, файл с решением должен называться* ***solution.c****. Результат работы про-*

граммы должен быть записан в файл ***result.txt****. Ваша программа должна обрабатывать директорию, которая называется* ***tmp****.*

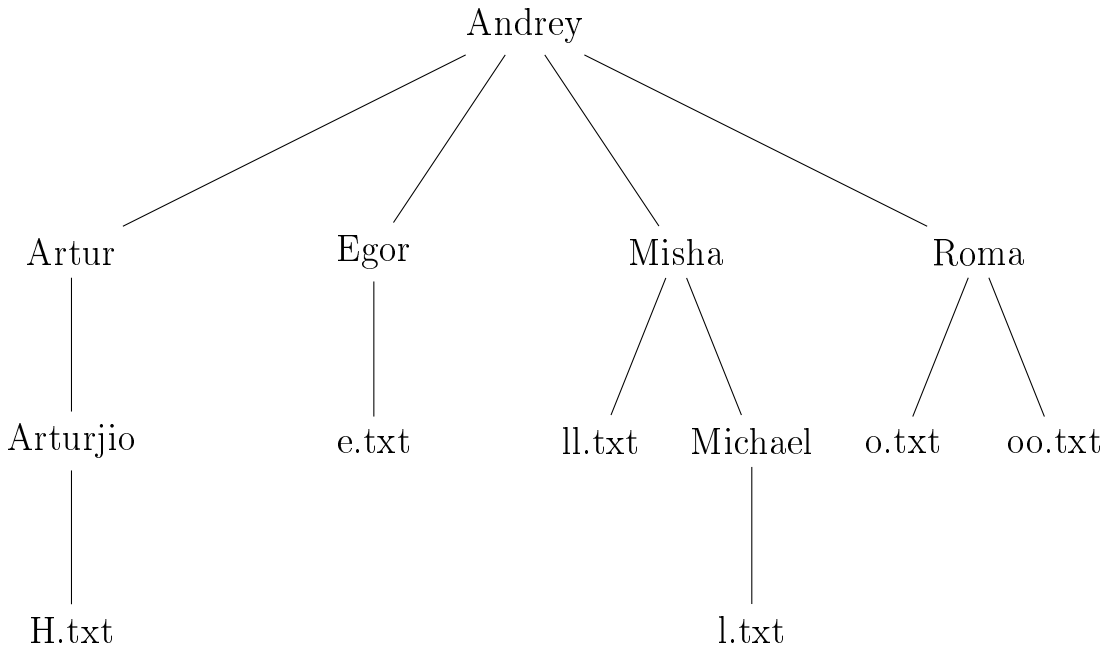
Выполнение работы:

В функции *main()* создается строка *char s[]="./tmp"* - это директория, из которой наша функция начнет работу. Затем, так как строка, которая надо найти, имеет неизвестную длину, происходит считывание с динамическим выделением памяти под эту строку. Следом мы создаем строку *char temp[]="1.txt"* - это строка, с помощью которой будет происходить поиск. Мы открываем файл result.txt на запись. Если файла не существует, он будет создан. Затем циклом от каждой нужной буквы мы запускаем рекурсивный обход файловой системы.

Рекурсивный обход происходит с помощью функции *list_dir(const char* dirPath,const char* need, FILE* f)*. В ней происходит открытие директории, имя которой подается на вход. Затем просматриваются все поддиректории, содержащиеся в этой директории. Если мы не нашли не директорию, а файл, подходящий по условию (условно H.txt), то путь к нему записывается в result.txt. Если же мы нашли новую директорию, то из нее рекурсивно запускается list_dir(), и опять происходит происходит просматривание файлов, содержащихся теперь в этой директории. Так мы обойдем все директории и поддиректории.

Тестирование:

Тестирование происходило на таком дереве файлов с исходной строкой "Hello":



Результат

```
/home/evildre/Рабочий стол/Программирование/Лаб3/Andrey/Artur/Arturjio/H.txt
/home/evildre/Рабочий стол/Программирование/Лаб3/Andrey/Egor/e.txt
/home/evildre/Рабочий стол/Программирование/Лаб3/Andrey/Misha/Michael/l.txt
/home/evildre/Рабочий стол/Программирование/Лаб3/Andrey/Misha/Michael/l.txt
/home/evildre/Рабочий стол/Программирование/Лаб3/Andrey/Roma/o.txt
```

Комментарий

Верно.

Вывод:

Была написана программа для работы с файловой и её рекурсивного обхода.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: solution.c

```
#include <stdio.h>
#include <dirent.h>
#include <stdlib.h>
#include <string.h>
void list_dir(const char* dirPath,const char* need, FILE* f)
{
        DIR* dir = opendir(dirPath);
        if (dir){
                struct dirent *de = readdir(dir);
                while (de){
                        if ((strcmp(de->d_name,".")!=0) && (strcmp(de->d_name,"..")!=0)){
                                char s[strlen(dirPath)+2+strlen(de->d_name)];
                                strcpy(s,dirPath);
                                strcat(s,"/");
                                strcat(s,de->d_name);
                                if(strcmp(de->d_name, need)==0){
                                        fprintf(f,"%s\n",s);
                                        return;
                                }
                                list_dir(s,need,f);
                                de = readdir(dir);
                        }
                        else{
                                de = readdir(dir);
                        }
                }
        }
        closedir(dir);
        return;
}
int main()
{
        char s[]="./tmp";
        char c='1';
        int k=0;
        char* need;
        need=malloc(sizeof(char));
        while(c!='\n'){
                scanf("%c",&c);
                k+=1;
                need=(char*)realloc(need,sizeof(char)*k);
                need[k-1]=c;
        }
        need[k]='\0';
        char temp[]="1.txt";
        FILE *file = fopen("result.txt","w");
        int i;
        for(i=0;i<strlen(need);i++){
                temp[0]=need[i];
                list_dir(s,temp,file);
        }
        free(need);
        fclose(file);
}
```