

Міністерство освіти і науки України
Національний технічний університет
«Харківський політехнічний інститут»
Кафедра «Обчислювальна техніка та програмування»

ЗВІТ
Про виконання лабораторної роботи №14
«Взаємодія з файлами»

Виконавець:
студент гр. КІТ-120В
Олексієнко Микита

Харків 2021

Лабораторна робота №14. Взаємодія з файлами

1.Вимоги

1.1 Розробник

Олексієнко Микита Віталійович

студент групи КІТ-120В

17.01.2021

1.2 Загальне завдання

Реалізувати програму відповідно до індивідуального завдання.

1.3 Індивідуальне завдання

Визначити об'єм запитаного каталогу.

2. Виконання роботи

2.1 Функціональне призначення

Програма призначена для визначення розміру запитаного каталогу.

2.2 Створення репозиторію

Створивши репозиторій на github, клоную його та створюю всі файли за вимогами структури проекту.

2.3 Написання коду програми

```
/**
```

```
 * @mainpage
```

```
 * # Загальне завдання
```

```
 * 3. **Визначити** об'єм запитаного каталогу
```

```
 * @author Alekseenko N.
```

```
 * @date 17-jan-2021
```

```
 * @version 1.0
```

```
 */
```

```
/**
```

```
 * @file main.c
```

```
* @brief Файл з демонстрацією роботи структури та методів
```

```
* оперування ним.
```

```
*
```

```
* @author Alekseenko N.
```

```
* @date 17-jan-2021
```

```
* @version 1.0
```

```
*/
```

```
#include <stdio.h>
```

```
#include <dirent.h>
```

```
#include <fcntl.h>
```

```
#include <sys/stat.h>
```

```
#include <stdint.h>
```

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

```
#include <string.h>
```

```
//static // статическая переменная для  
определения размерности директории
```

```
/**
```

```
* *Функція tree_result*
```

```
*
```

```
* Послідовність дій:
```

```
* - оголошення змінних
```

```
* \param mb_size - розмір директорії
```

```
* \param end - параметр для очищення адреси файлу
```

```
* \param struct stat statbuf - структура stat
```

```
* \param struct dirent *dp - структура dirent
```

```
* \param DIR *d - показник на назву директорії
```

```
* - Спочатку вписуємо назву директорії, яку потрібно прочитати
```

```
* - Перевіряємо можливість відкриття директорії
```

* - Рахуємо об'єм директорії і передаємо його в основну функцію для виведення результату

* @return об'єм директорії (mb_size)

*/

double tree_result(char *t, int l)

{

double mb_size = 0.0;

size_t end;

struct stat statbuf;

DIR *d = opendir(t);

struct dirent *dp;

int ddd, fff;

if ((d = fdopendir((ddd = open(t, O_RDONLY)))) == NULL) // вариант
невозможности открытия директории

{

fprintf(stderr, "Cannot open %s directory\n", t);

return 1;

}

while ((dp = readdir(d)) != NULL) // цикл для
считывания содержимого директории

{

if (dp->d_name[0] == '.')

continue;

if ((fff = openat(ddd, dp->d_name, O_RDONLY)) == -1)

{

perror(dp->d_name);

continue;

}

if (fstat(fff, &statbuf) == 0 && (statbuf.st_mode & S_IFDIR))

{

end = strlen(t);

```
strcat ( t, "/" ); //
```

образование адреса файла

```
strcat ( t, dp->d_name );
```

```
mb_size += tree_result( t, l + 1 );
```

```
t[end] = 0;
```

```
}
```

```
else
```

```
{
```

```
mb_size += ( double ) statbuf.st_size / 1000 ; //
```

вычисляем размерность директории

```
}
```

```
close(fff);
```

```
}
```

```
closedir(d);
```

```
return mb_size;
```

```
}
```

```
/**
```

```
* *Функція main*
```

```
*
```

```
* Послідовність дій:
```

```
* - оголошення змінних
```

```
* \param text - вхідна назва папки
```

```
* @return успішний код повернення з програми (0)
```

```
*/
```

```
int main ()
```

```
{
```

```
char text[100];
```

```
scanf("%s", text );
```

```
printf("\nSize of the '%s' directory is: %.2f kB\n", text, tree_result(text, 0));
```

```
return 0;
```

```
}
```

2.4 Перевірка правильності роботи програми.

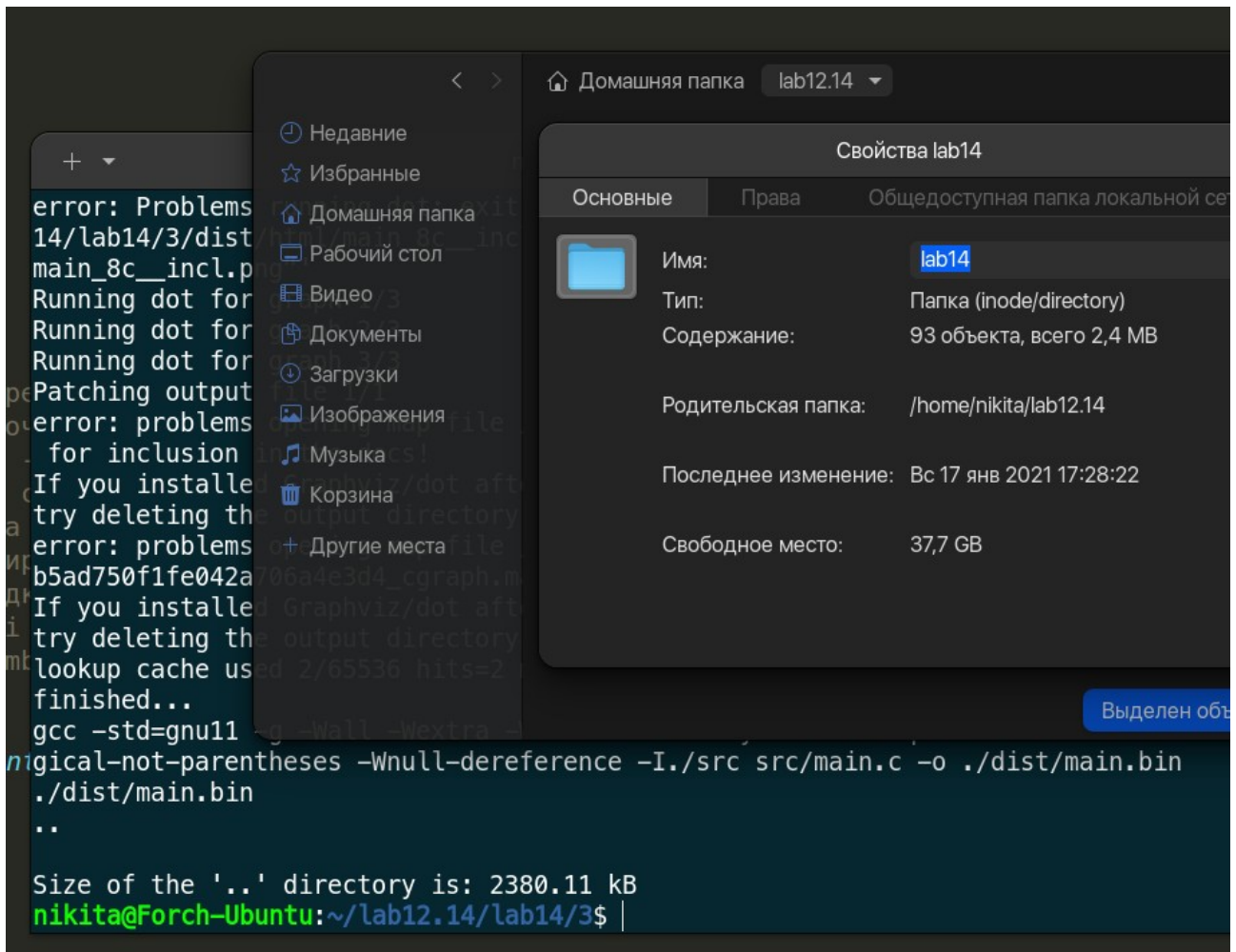


Рисунок 1 — Результат

2.5 Для поліпшення сприйняття коду, я розробив блок-схему:

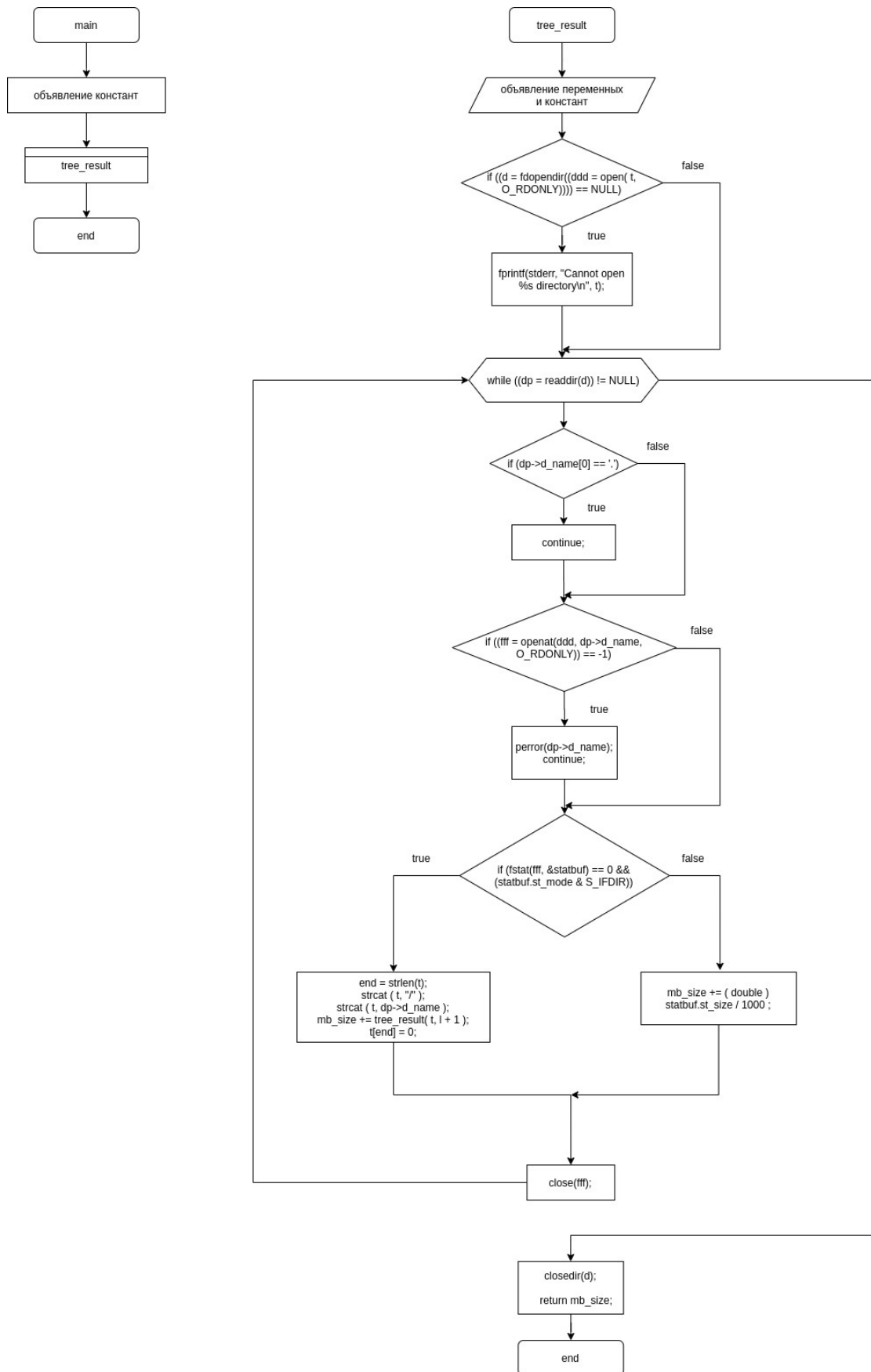


Рисунок 2 — Блок-схема

2.6 Опис функцій:

Опис функцій

◆ main()

```
int main ( )
```

Функція *main*

Послідовність дій:

- оголошення змінних

Аргументи

text - вхідна назва папки

Повертає

успішний код повернення з програми (0)

Граф всіх викликів цієї функції:

◆ tree_result()

```
double tree_result ( char * t,  
                    int    l  
                    )
```

Функція *tree_result*

Послідовність дій:

- оголошення змінних

Аргументи

mb_size - розмір директорії

end - параметр для очищення адреси файлу

struct stat statbuf - структура stat

struct dirent *dp - структура dirent

DIR *d - показчик на назву директорії

- Спочатку вписуємо назву директорії, яку потрібно прочитати
- Перевіряємо можливість відкриття директорії
- Рахуємо об'єм директорії і передаємо його в основну функцію для виведення результату

Повертає

об'єм директорії (mb_size)

Рисунок 3 — Опис функцій

Висновок:

Я створив програму, яка по введеної папці показує її розмір.