# Лабораторна робота №13. Взаємодія з файлами

## 1 ВИМОГИ

### 1.1 Розробник

* Хелемендик Дмитро Олегович;
* студент групи КІТ-121д;
* 13-січ-2022.

### 1.2 Загальне завдання

Вивести структуру файлів та каталогів, як це робить утиліта Linux tree. Виклик функції system заборонено.

## 2 ОПИС ПРОГРАМИ

### 2.1 Функціональне призначення:

Програма призначена для виводу всіх файлів та каталогів у заданої директорії та запис цих даних в заданий файл, як це робить утиліта Linux tree. Програма працює за допомогою функцій, що задекларовані в *lib.h, stdio.h, dirent.h.*

Результат зберігається у заданому файлі. Наприклад, output.txt.

Демонстрація знайдених результатів передбачає як покрокове виконання програми в режимі налагодження, так і видача даних у вікні консолі.

### 2.2 Опис логічної структури

Опис розроблених структур і функцій наводиться на базі результатів роботи системи автодокументування *Doxygen*.

#### Функція отримання розміру файла, з якого будуть зчитані дані

int size\_of\_file(char \*argv[]);

*Призначення*: отримання розміру файла.

*Опис роботи*: функція визначає розмір файла з вхідними даними.

**Аргументи:**

- *argv* - масив, котрий зберігає введені користувачем дані;

#### Функція отримання вхідних даних

void read\_from\_file(char \*buff, char \*argv[]);

*Призначення*: отримання вхідних даних.

*Опис роботи*: функція зчитує вхідні дані та записує їх в буфер.

**Аргументи:**

- *buff* - буфер;

- *argv* - масив, котрий зберігає введені користувачем дані;

#### Функція отримання даних заданої директорії та запис отриманних даних у файл

void open\_dir\_and\_write\_output(int argc, char \*argv[]);

*Призначення*: отримання структуру файлів та каталогів.

*Опис роботи*: функція відкриває задану директорію, читає її, записує дані в заданий файл та друкує ці дані в командній строці.

**Аргументи:**

- *argc* - кількість заданих аргументів користувачем;

- *argv* - масив, котрий зберігає введені користувачем дані;

#### Основна функція

int main()

*Призначення*: головна функція.

*Опис роботи*:

- роблю перевірку на кількість аргументів введених користувачем;

- знаходю розмір файла, з якого будуть зчитані дані функцією {@link size\_of\_file};

- виділяю пам'ять для буфера;

- зчитую дані з файлу та записую ці дані в буфер шляхом виклика функції {@link read\_from\_file};

- виконую функцію {@link open\_dir\_and\_write\_output}, яка записує файли заданої директорії в заданий файл;

- звільнюю пам'ять;

- успішний код повернення з програми (0).

#### Структура проекту:

└── lab13

├── assets

│ └── input.txt

├── doc

│ ├── lab13.docx

│ └── lab13.md

│ └── lab13.pdf

├── Doxyfile

├── Makefile

├── README.md

├── src

│ ├── lib.c

│ ├── lib.h

│ └── main.c

└── test

└── test.c

#### 2.3 Важливі фрагменти програми

#### Знаходження розміру файла з вхідними файлами

FILE \*f = fopen(argv[1], "r");

if (!f) {

perror("fopen");

}

fseek(f, 0L, SEEK\_END);

int size = (int)ftell(f);

fseek(f, 0L, SEEK\_SET);

fclose(f);

#### Запис даних в заданий файл та їхній друк у консоль

fprintf(f, "In %s directory we have this file(s):\n", argv[1]);

while ((ent = readdir(dir)) != NULL) {

fprintf(f, "%s\n", ent->d\_name);

printf("%s\n", ent->d\_name);

}

#### 3. Варіанти використання

Для демонстрації результатів кожної задачі використовується:

- покрокове виконання програми в утиліті lldb;

- виконання програми у вікні консолі.

**Варіант використання 1**: послідовність дій для запуску програми у режимі відлагодження:

- запустити програму у відлагоднику lldb з трьома аргументами(перший - бінарний файл, другий - задана директорія, третій - файл, у який буде записаний результат);

- поставити точку зупинки на функції main (строка з `return 0;`);

- запустити програму;

- подивитися результат виконання програми.

Також результат зберігається у заданому файлі(зараз це output.txt).

dima@dima-VirtualBox:~/dev/programing-khelemendyk/lab13/dist$ lldb main.bin "/home/dima/dev/programing-khelemendyk/lab13/assets/input.txt" "output.txt"

(lldb) b 48

(lldb) r

README.md

Doxyfile

src

dist

Makefile

test

.

..

doc

Process 7933 stopped

\* thread #1, name = 'main.bin', stop reason = breakpoint 1.1

frame #0: 0x00000000004012c2 main.bin`main(argc=3, argv=0x00007fffffffe068) at main.c:35:2

32 int main(int argc, char \*argv[])

33 {

34 open\_dir\_and\_write\_output(argc, argv);

-> 35 return 0;

36 }

**Варіант використання 2**: запуск програми у вікні консолі:

- запустити програму у консолі з трьома аргументами;

- подивитись результат програми.

Також результат зберігається у заданому файлі(зараз це output.txt).

dima@dima-VirtualBox:~/dev/programing-khelemendyk/lab13/dist$ ./main.bin "/home/dima/dev/programing-khelemendyk/lab13/assets/input.txt" "output.txt"

README.md

Doxyfile

src

dist

Makefile

test

.

..

doc

## ВИСНОВКИ

При виконанні даної лабораторної роботи було набуто практичного досвіду у взаємодії з файлами.