

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

ЗВІТ

Про виконання лабораторної роботи № 12, 14
«Взаємодія з користувачем шляхом механізму введення/виведення.
Взаємодія з файлами»

Керівник: викладач
Бульба С. С.

Виконавець: студент гр. КІТ-120в
Бабенко А. П.

Харків 2021

Лабораторна робота №12, 14. Взаємодія з користувачем шляхом механізму введення/виведення. Взаємодія з файлами

1. Вимоги

1.1 Розробник

- Бабенко Антон Павлович;
- Студент групи КІТ-120в;
- 12 лютого 2021;

1.2 Загальне завдання

На оцінку “відмінно”. (Треба переробити лише одну програму на вибір з попередньої роботи). Треба продемонструвати взаємодію з користувачем шляхом використання функцій:

- *printf()* та *scanf()*;
- *gets()*, *getc()* та *puts()*, *putc()*;
- *write()*, *read()*

На оцінку “відмінно”. Виконати усі завдання з переліку:

1. Вивести структуру файлів та каталогів, як це робить утиліта Linux *tree*. Виклик функції *system* заборонено.
2. Визначити об'єм запитаного каталогу. Результат нормалізувати. Формат розміру:
 - не більше ніж 3 знаки до коми;
 - не більше ніж 2 знаки після коми;
 - єдиний випадок, коли в чисельній частині числа може бути 0 - коли розмір дорівнює 0 байт.

1.3 Індивідуальне завдання

Програму, яка була розроблена у попередній лабораторній роботі (робота з показниками), змінити так, щоб:

- початкові дані вводилися з клавіатури;
- видача результуючих даних провадилася у консоль.
- при старті програми виводилась інформація об авторі, номері та темі лабораторної роботи;
- при запиті даних, користувач отримав повідомлення, що від нього очікують.

Програму, яка була розроблена у попередній лабораторній роботі (робота з строками), змінити так, щоб:

- початкові дані вводилися з файлу;
- видача результуючих даних провадилася не тільки у консоль, але й у файл.
- ім'я вхідного та результуючого фалу повинно отримано від користувача;
- при запиті даних, користувач отримав повідомлення, що від нього очікують.

1.4 Функціональне призначення

Програма №1 призначена для пошуку чисел які повторюються і кількості повторень. Програма №2 призначена для виведення структури директорій і файлів.

2 Виконання роботи

2.1 Написання коду, що демонструє взаємодію з користувачем . Зображено на рис.1.

```
puts("\nЛабораторна робота №12");
write(1, "Взаємодія з користувачем шляхом механізму введення/виведення\n", 115);
printf("\nEnter author's surname\n");
gets(surname);
printf("\nEnter first letter of author's name\n");

char name =getc(stdin);
printf("\nAuthor: %s %c\n", surname, name);

printf("\nEnter 0 (start) or 1 (exit)\n");
char isEnterCh;
read(0, &isEnterCh, 1);
int isEnter = isEnterCh - '0';

if (isEnter != 0) {
    printf("You entered ");
    putc(isEnter + '0', stdout);
    printf(", program closed\n");
    exit(0);
}

int N = 0;
printf("\nEnter count of numbers\n");
scanf("%d", &N);
```

Рисунок 1 – код програми

2.2 Виконання опису функцій. Зображення опису зі сторінок документації doxygen на рис. 2.

◆ main()

int main ()

Головна функціяcd.

Послідовність дій:

- Взаємодія з користувачем шляхом механізму введення/виведення
- задання кількості елементів масиву
- ініціалізація масиву
- виклик функції fill
- виклик функції repeat

Повертає
успішний код виконання програми (0)

Рисунок 2 – опис функції

2.3 Ставимо точку зупину, проходимо по файлу і бачимо зміну значення в масиві в момент проходження по програмі. Зображено на рис.3.

```
printf("\nEnter first letter of author's name\n");

char name = getc(stdin);
printf("\nAuthor: %s %c\n", surname, name);

printf("\nEnter 0 (start) or 1 (exit)\n");
```

переменная	Значение	Тип
Локальные переменные		
name	65 'A'	char

Рисунок 3 – файл у відлагоднику

2.4 Створення блок-схеми програми. Зображення блок-схеми на рис.4.

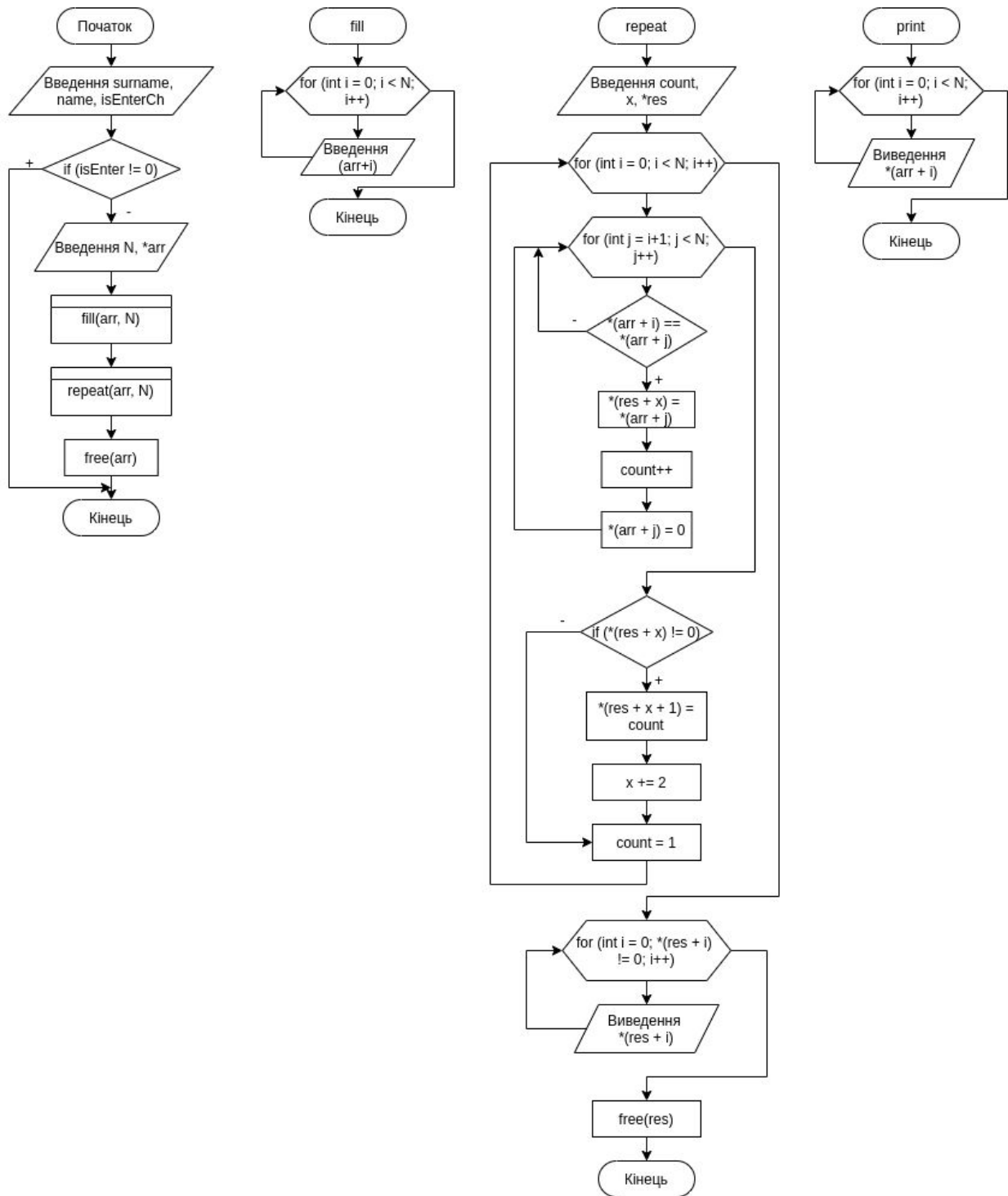


Рисунок 4 – блок-схема

2.5 Написання коду, що виводить структуру файлів та каталогів . Зображено на рис.5.

```

void tree(char *path, int root)
{
    char newPath[N];
    DIR *dir;
    struct dirent *dp;
    if ((dir = opendir(path)) != NULL)
    {
        while ((dp = readdir(dir)) != NULL)
        {
            if (strcmp(dp->d_name, ".") != 0 && strcmp(dp->d_name, "..") != 0)
            {
                /* Вывод символов для отступа */
                for (int i = 0; i < root; i++)
                {
                    if (i % 2 == 0 || i == 0)
                        printf("|");
                    else
                        printf(" ");
                }

                printf("├─%s\n", dp->d_name);

                /* Создание нового пути из исходного */
                strcpy(newPath, path);
                strcat(newPath, "/");
                strcat(newPath, dp->d_name);

                tree(newPath, root + 2);
            }
        }
    }
}

```

Рисунок 5 – код програми

2.6 Виконання опису функцій. Зображення опису зі сторінок документації doxygen на рис. 6.



Головна функція

Послідовність дій:

- ініціалізація строки для запису шляху до директорії
- зчитування шляху від користувача
- виклик функції **tree()**

Повертає

успішний код виконання програми (0)

Рисунок 6 – опис функції

2.7 Ставимо точку зупину, проходимо по файлу і бачимо зміну значення в масиві в момент проходження по програмі. Зображено на рис.7.

▼ Локальные переменные		
▼ path	[1000]	char [1000]
0	47 '/'	char
1	104 'h'	char
2	111 'o'	char
3	109 'm'	char
4	101 'e'	char
5	47 '/'	char

Рисунок 7 – файл у відлагоднику

2.8 Створення блок-схеми програми. Зображення блок-схеми на рис.8.

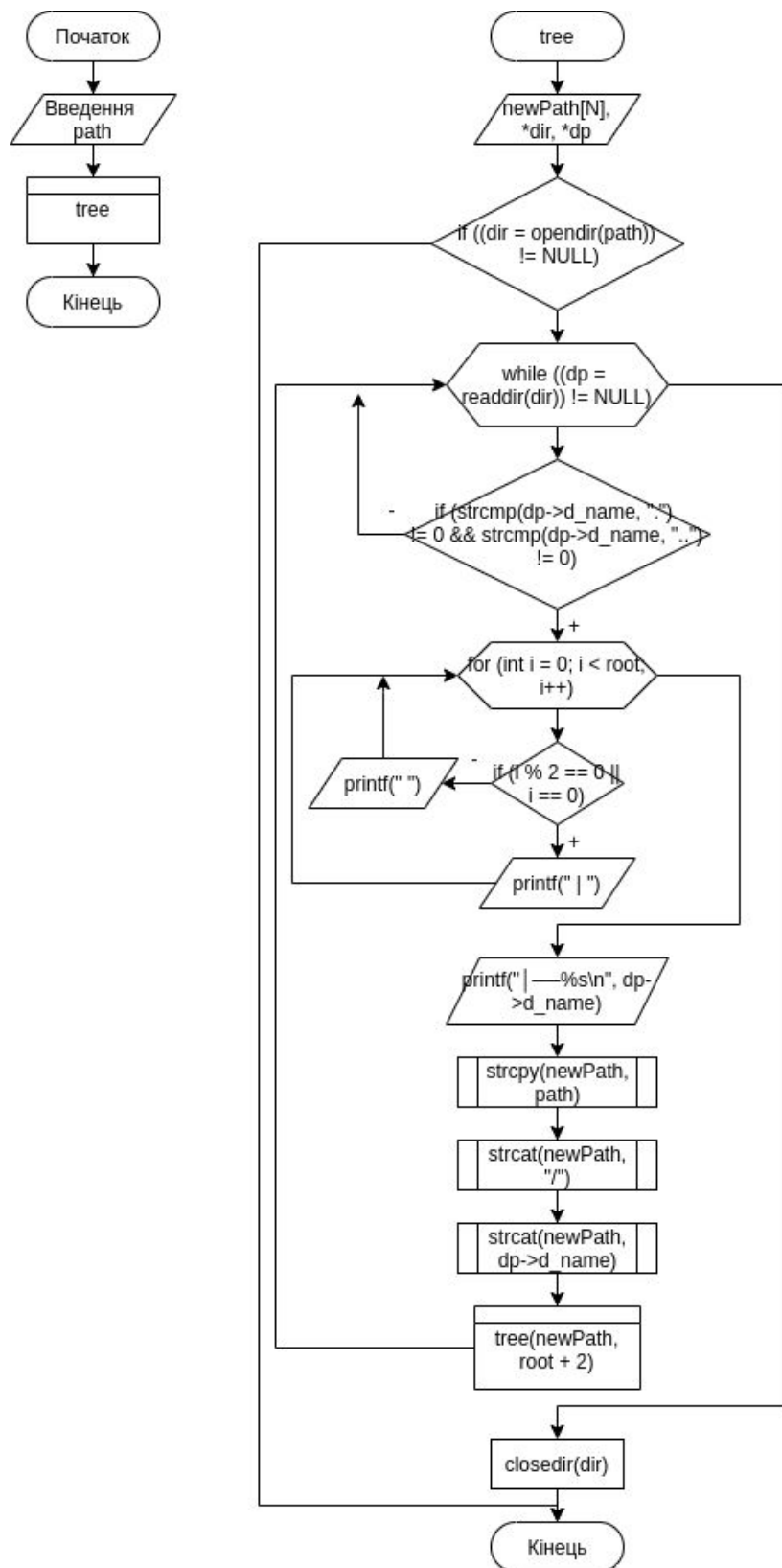


Рисунок 8 – блок-схема

Висновки

При виконанні даної лабораторної роботи було набуто навичок розробки програм для взаємодії з користувачем та для взаємодії з файлами.