

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

ЗВІТ

Про виконання лабораторної роботи № 15, 17
«Структуровані типи даних. Модульні тести»

Керівник: викладач
Бульба С. С.

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

Харків 2021

Лабораторна робота №15, 17. Структуровані типи даних. Модульні тести

1. Вимоги

1.1 Розробник

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

1.2 Загальне завдання

- з розділу “Розрахункове завдання / Індивідуальні завдання” взяти прикладну галузь стосовно номеру варіанту за попередньо-визначеною формулою
- створити структуру, що відображає “базовий клас”

1.3 Індивідуальне завдання

На оцінку “відмінно”.

1. Виконати завдання на оцінку “задовільно” та “добре” згідно варіанту.
2. Виконати запис та читання масиву структур (з заздалегідь відомою кількістю елементів) у двох форматів: текстовому та бінарному.
3. При бінарному методі зберігання структур виконати пошук та читання структури з файлу по індексу.

1.4 Функціональне призначення

Програма №1 призначена для запису і читання масиву структур у бінарному форматі. Програма №2 призначена для пошуку та читання структури з файлу по індексу при бінарному методі.

2 Виконання роботи

2.1 Написання коду, що шукає структуру за індексом . Зображено на рис.1.

```
void find_struct(char *buff, file *arr, int N, int number) {
    FILE *fp = fopen(buff, "rb");
    file founded_struct;
    fseek(fp, number*sizeof(file), SEEK_SET);
    fread(&founded_struct, sizeof(file), 1, fp);
    fclose(fp);
    printf("\n\n\n\n");

    printf("Visibility: %s\n", bool_output(founded_struct.isVisible));
    printf("Filename: %s\n", founded_struct.filename);
    printf("File size: %.3f kb\n", founded_struct.size);
    printf("Readable: %s\n", bool_output(founded_struct.access.read));
    printf("Writable: %s\n", bool_output(founded_struct.access.write));
    printf("Executable: %s\n", bool_output(founded_struct.access.execute));
    printf("File extension: %s\n", founded_struct.extension);
};
```

Рисунок 1 – код програми

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

◆ main()

int main ()

Головна функція

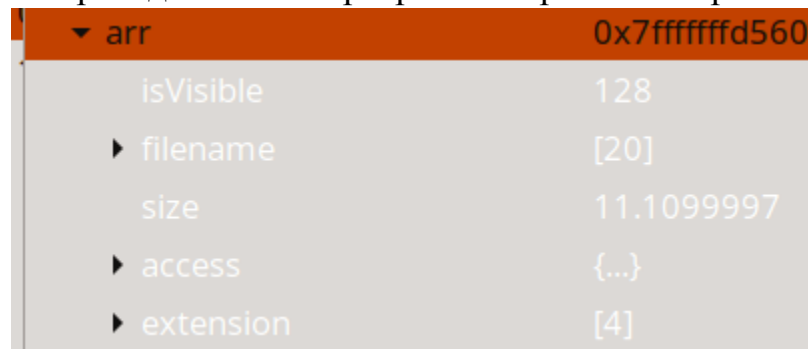
Послідовність дій:

- зчитування кількості строк у файлі
- ініціалізація масиву структур
- виклик функції `read_from_file()`
- виклик функції `write_to_file_bin()`
- виклик функції `read_from_file_bin()`
- виклик функції `output()`
- ініціалізація номеру структури для пошуку
- виклик функції `find_struct()`

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.



▼ arr	0x7fffffff560
isVisible	128
▶ filename	[20]
size	11.1099997
▶ access	{...}
▶ extension	[4]

Рисунок 3 – файл у відлагоднику

2.4 Створення блок-схеми програми і графу викликів функції main. Зображення блок-схеми на рис.4-5.

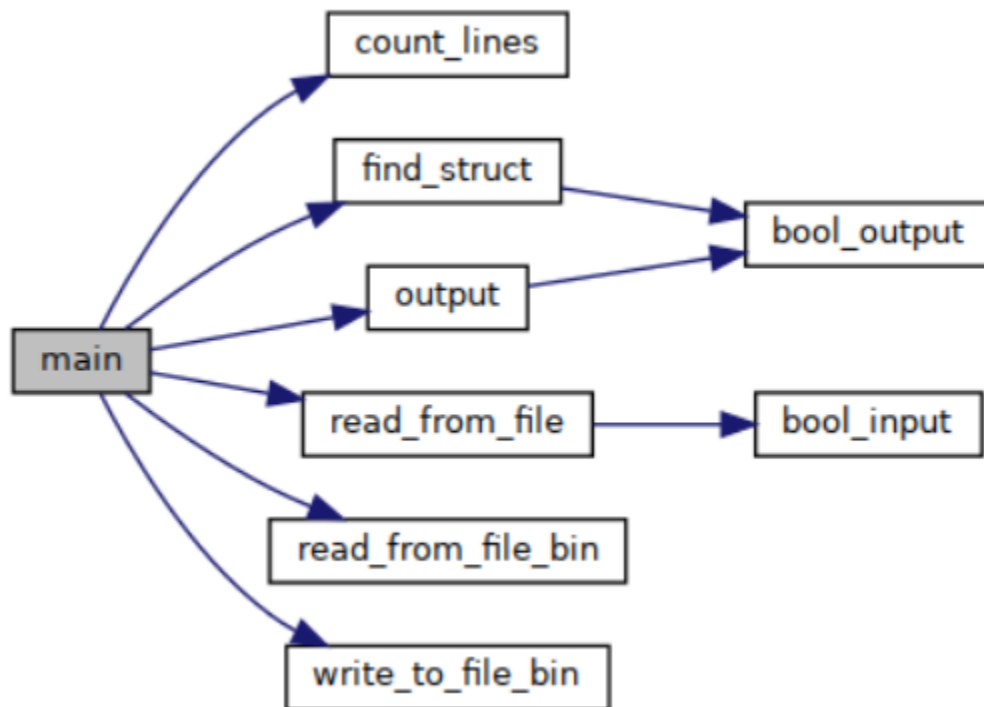


Рисунок 4 – граф викликів функції

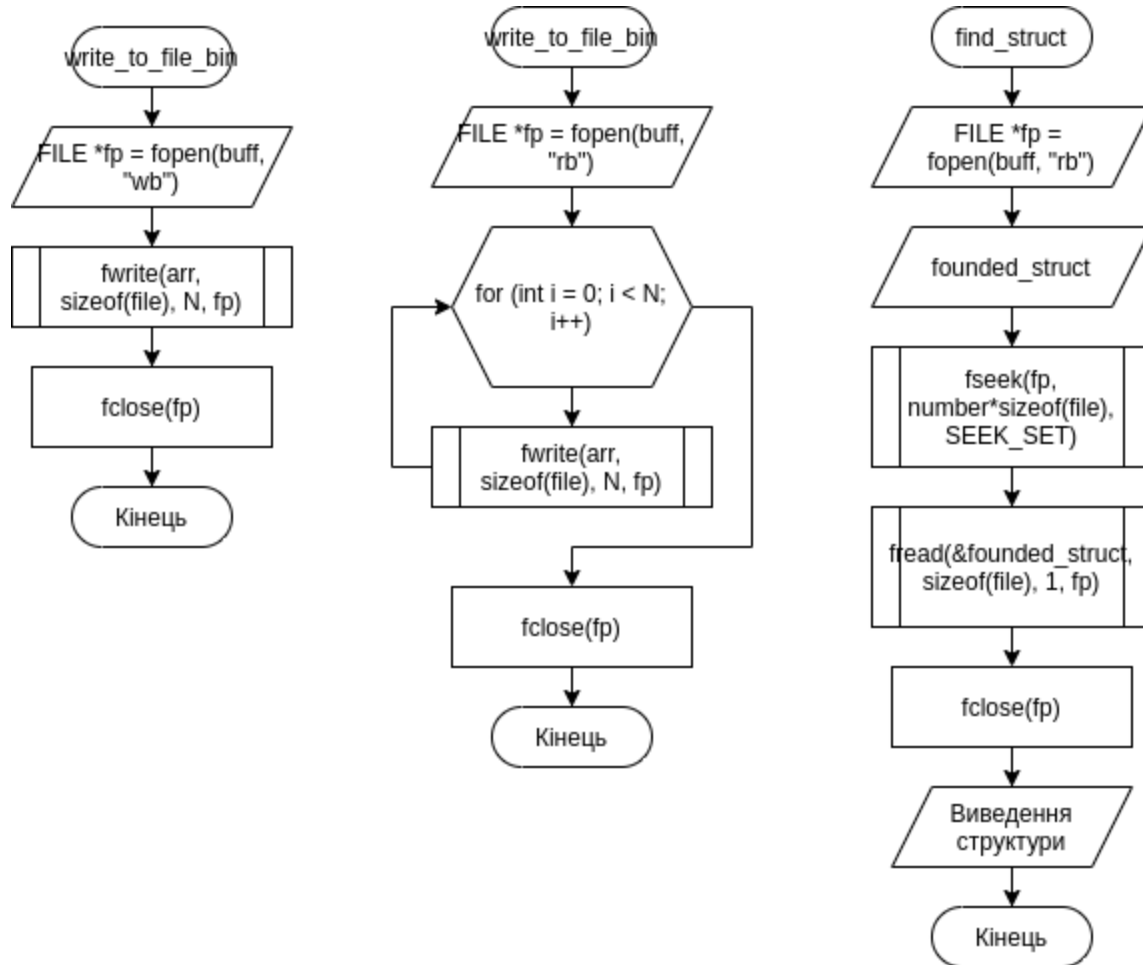


Рисунок 5 – блок-схема

2.5 Написання коду для тестування функції згідно індивідуального варіанту. Зображено на рис.6.

```

unsigned short test_pick_by_size()
{
    unsigned short is_success = 1;
    file testFiles[10];
    read_from_file(testFiles, 10, "test.txt");
    int N = 6;
    int K = pick_by_size(testFiles, 10, 50);
    if (N != K) {
        printf("\nThe number of files > 50 kb was expected to be %d, but the function returned: %d\n", N, K);
        is_success = 0;
    }
    return is_success;
}
  
```

Рисунок 6 – код тесту

2.6 Перевірка правильності роботи функції за допомогою тесту. Зображено на рис.7.

```
Name of file > 50 kb: file5  
  
Name of file > 50 kb: file6  
  
Name of file > 50 kb: file7  
  
Name of file > 50 kb: file8  
  
Name of file > 50 kb: file9  
  
Name of file > 50 kb: file10  
  
Congratulations! All tests succeed!
```

Рисунок 7 – виведення тесту

Висновки

При виконанні даної лабораторної роботи було набуто навичок розробки програм зі структурованими типами даних і написання тестів.