

Лабораторна робота №14. Структуровані типи даних

1 ВИМОГИ

1.1 Розробник

- Хелемендик Дмитро Олегович;
- студент групи КІТ-121д;
- 14-квіт-2022.

1.2 Загальне завдання

З розділу “Індивідуальні завдання комплексної роботи” взяти прикладну галузь стосовно номеру варіанту за попередньо-визначеною формулою. Створити структуру, що відображає “базовий клас”.

2 ОПИС ПРОГРАМИ

2.1 Функціональне призначення:

Програма призначена для знаходження згорівших лампочок за допомогою зчитування даних з файла та сортування за заданим критерієм. Програма працює за допомогою функцій, що задекларовані *lib.h*, *stdlib.h*, *string.h* та *stdio.h*.

Результат зберігається у змінній *bulbs*.

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

2.2 Опис логічної структури

За допомогою ключового слова *struct* описуємо лампочку, що має 8 полів – чи ввімкнена лампочка, чи перегоріла лампочка, виробник, зворотній лічильник, ватти, температура колбору світіння, форма, тип цоколю.

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

Розроблено структуру, вміст якої подано нижче.

bulb
+ is_on + is_burn + factory + reverse_cntr + vatt + temp + form + type_plinth

Рисунок 1 — поля структури bulb

Функція заповнення структури

```
int write_to_struct(char *string, const char *delim, struct Bulb *bulbs);
```

Призначення: заповнить структуру даними.

Опис роботи: функція розбиває строку на частини та записує дані у структуру.

Аргументи:

- *string* — строка;
- *delim* — роздільник;
- *bulbs* — показчик на структуру лампочки.

Функція підрахунку кількості строк

```
int count_lines(char *argv[]);
```

Призначення: отримання кількості строк для визначення кількості лампочок.

Опис роботи: функція рахує кількість строк у файлі.

Аргументи:

- *argv* - масив з аргументи(введеними користувачем).

Функція знаходження найдовшої строки

```
int max_string_length(char *argv[]);
```

Призначення: знаходження найдовшої строки для визначення розміру буферу.

Опис роботи: функція знаходе кількість символів у найдовшій строці.

Аргументи:

- *argv* - масив з аргументи(введеними користувачем).

Функція заповнення структур та отримання критерія для сортування

```
int get_struct_and_criterion(struct Bulb *bulbs, int number_bulbs, char *argv[],  
char *criterion_for_sorting);
```

Призначення: заповнення структур та отримання критерія для сортування.

Опис роботи: функція зчитує дані з файла, заповнює масив структур та записує у показчик критерій для сортування за його наявності.

Аргументи:

- *bulbs* - показчик на структуру;

- *criterion_for_sorting* - кількість лампочок;

- *argv* - масив з аргументи(введеними користувачем);

- *criterion_for_sorting* - показчик для критерія сортування.

Функція знаходження згорівшої лампочки

```
int is_burned_bulbs(struct Bulb *bulbs, int number_bulbs);
```

Призначення: визначити кількість згорівших лампочок.

Опис роботи: функція знаходе номери згорівших лампочок та записує їх у показчик.

Аргументи:

- *bulbs* - показчик на структуру;

- *number_bulbs* - кількість лампочок.

Функція запису номера згорівшої лампочки

```
void get_index_burned_bulbs(struct Bulb *bulbs, int number_bulbs, int  
*burned_bulbs);
```

Призначення: записати номер згорівшої лампочки у показчик.

Опис роботи: функція знаходе номера згорівших лампочок та записує їх у показчик.

Аргументи:

- *bulbs* - показчик на структуру;
- *number_bulbs* - кількість лампочок;
- *burned_bulbs* - показчик для зберігання номерів згорівших лампочок.

Функція друку результату у консоль

```
void print_res_screen(struct Bulb *bulbs, int number_bulbs, int *burned_bulbs,  
int number_burned_bulbs, char *criterion_for_sorting);
```

Призначення: друк результату у вікно консолі.

Опис роботи: функція друкує результат у вікно консолі.

Аргументи:

- *bulbs* - показчик на структуру;
- *number_bulbs* - кількість лампочок;
- *burned_bulbs* - показчик для зберігання номерів згорівших лампочок;
- *number_burned_bulbs* - кількість згорівших лампочок;
- *criterion_for_sorting* - показчик для критерія сортування.

Функція друку результату у файл

```
void print_res_file(struct Bulb *bulbs, int number_bulbs, int *burned_bulbs, int  
number_burned_bulbs, char *criterion_for_sorting, char *argv[]);
```

Призначення: друк результату у файл.

Опис роботи: функція друкує результат у файл.

Аргументи:

- *bulbs* - показчик на структуру;
- *number_bulbs* - кількість лампочок;
- *burned_bulbs* - показчик для зберігання номерів згорівших лампочок;
- *number_burned_bulbs* - кількість згорівших лампочок;
- *criterion_for_sorting* - показчик для критерія сортування;
- *argv* - масив з аргументи(введеними користувачем).

Основна функція

```
int main()
```

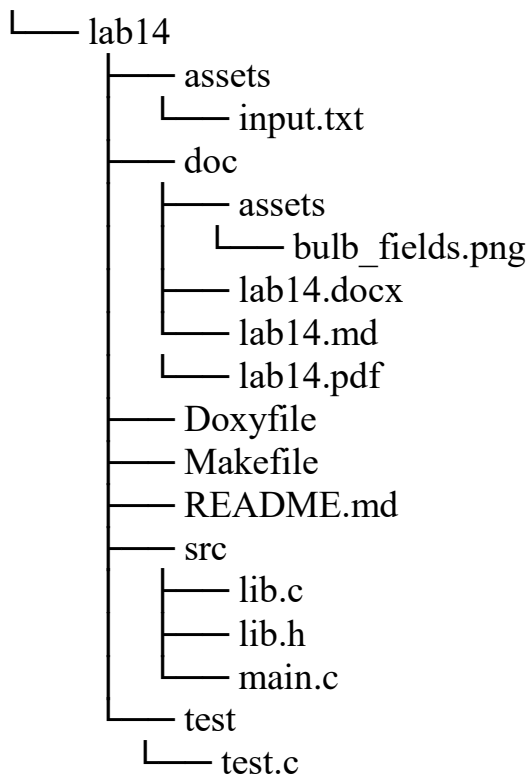
Призначення: головна функція.

Опис роботи:

- знаходю кількість лампочок за допомогою функції *count_lines* та зберігаю дані у змінній *number_bulbs*;
- виділяю пам'ять для масиву символів, у якому буде зберігатися критерій для сортування;
- виділяю пам'ять для структури розміром *number_bulbs*;
- далі зчитую дані з файлу, записую критерій(якщо він є), заповнюю структуру шляхом виклику функції *get_struct_and_criterion*;
- знаходю кількість згорівших лампочок функцією *is_burned_bulbs* та зберігаю у змінній *number_burned_bulbs*;
- виділяю пам'ять для масиву з номерами згорівших лампочок;
- знаходю номери згорівших лампочок та зберігаю їх в показчику *burned_bulbs*;
- друкую результат у вікно консолі за допомогою функції *print_res_screen*;

- якщо введений третій аргумент - друкую результат у файл шляхом виклику функції `print_res_file`;
- звільнюю пам'ять;
- успішний код повернення з програми (0).

Структура проекту:



2.3 Важливі фрагменти програми

Запис до структури

```
char *p = strtok(string, delim);
if (!p || !strncpy(bulbs->is_on, p, sizeof(bulbs->is_on) - 1))
    return 1;

if (!(p = strtok(NULL, delim)) || !strncpy(bulbs->is_burned, p, sizeof(bulbs->is_burned) - 1))
    return 1;
```

```

        if (!(p = strtok(NULL, delim)) || !strncpy(bulbs->factory, p, sizeof(bulbs->factory) - 1))
            return 1;
        if (!(p = strtok(NULL, delim)) || sscanf(p, "%d", &(bulbs->reverse_counter)) != 1)
            return 1;
        if (!(p = strtok(NULL, delim)) || sscanf(p, "%d", &(bulbs->vatt)) != 1)
            return 1;
        if (!(p = strtok(NULL, delim)) || sscanf(p, "%d", &(bulbs->color_temp)) != 1)
            return 1;
        if (!(p = strtok(NULL, delim)) || !strncpy(bulbs->shape, p, sizeof(bulbs->shape) - 1))
            return 1;
        if (!(p = strtok(NULL, delim)) || !strncpy(bulbs->base_type, p, sizeof(bulbs->base_type) - 1))
            return 1;

    return 0;

```

Заповнення структур та отримання критерію сортування

```

fgets(criterion_for_sorting, 50, f);
int size_buffer = max_string_length(argv);
char *buffer = (char *)malloc((unsigned int)size_buffer);
for (int i = 0; i < number_bulbs; i++) {
    fgets(buffer, size_buffer, f);
    if (write_to_struct(buffer, ",", (bulbs + i))) // заповнюю масив структур
        fprintf(stderr, "Error!\n");
}

```

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 - путь до файла, 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/lab14/dist$ lldb  
main.bin "/home/dima/dev/programing-khelemendyk/lab14/assets/input.txt"  
"output.txt"
```

```
(lldb) target create "main.bin"
```

```
Current executable set to '/home/dima/dev/programing-khelemendyk/lab14/dist/main.bin' (x86_64).
```

```
(lldb) settings set -- target.run-args "/home/dima/dev/programing-khelemendyk/lab14/assets/input.txt" "output.txt"
```

```
(lldb) b 69
```

```
Breakpoint 1: where = main.bin`main + 233 at main.c:69:2, address =  
0x0000000000402a49
```

```
(lldb) r
```

```
Process 5252 launched: '/home/dima/dev/programing-khelemendyk/lab14/dist/main.bin' (x86_64)
```

Your bulbs:

Bulb 1: yes, no, TOV Roga ta koputa, 20, 15, 1800, Globe, E40

Bulb 2: no, yes, TOV Roga ta oleni, 40, 30, 3600, Circle, E20
Bulb 3: no, no, TOV Koputa, 30, 25, 1900, Pear, E27
Bulb 4: yes, yes, TOV Romashka, 25, 150, 2500, Candle, E50
Bulb 5: yes, yes, TOV Kapysta, 19, 10, 1400, Ogive, E30

Burnt out bulbs:

Bulb 2: no, yes, TOV Roga ta oleni, 40, 30, 3600, Circle, E20
Bulb 4: yes, yes, TOV Romashka, 25, 150, 2500, Candle, E50
Bulb 5: yes, yes, TOV Kapysta, 19, 10, 1400, Ogive, E30

Your criterion for sorting: Bulb manufacturer

Bulb 1: TOV Roga ta koputa

Bulb 2: TOV Roga ta oleni

Bulb 3: TOV Koputa

Bulb 4: TOV Romashka

Bulb 5: TOV Kapysta

Process 5252 stopped

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

```
frame      #0:      0x0000000000402a49      main.bin`main(argc=3,
argv=0x00007ffffffe048) at main.c:69:2
   66          free(type_for_sort);
   67          free(burn_bulbs);
   68      }
-> 69      return 0;
   70 }
```

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/lab14/dist$  
./main.bin          "/home/dima/dev/programing-khelemendyk/lab14/assets/input.txt"  
"output.txt"
```

Your bulbs:

Bulb 1: yes, no, TOV Roga ta koputa, 20, 15, 1800, Globe, E40

Bulb 2: no, yes, TOV Roga ta oleni, 40, 30, 3600, Circle, E20

Bulb 3: no, no, TOV Koputa, 30, 25, 1900, Pear, E27

Bulb 4: yes, yes, TOV Romashka, 25, 150, 2500, Candle, E50

Bulb 5: yes, yes, TOV Kapysta, 19, 10, 1400, Ogive, E30

Burnt out bulbs:

Bulb 2: no, yes, TOV Roga ta oleni, 40, 30, 3600, Circle, E20

Bulb 4: yes, yes, TOV Romashka, 25, 150, 2500, Candle, E50

Bulb 5: yes, yes, TOV Kapysta, 19, 10, 1400, Ogive, E30

Your criterion for sorting: Bulb manufacturer

Bulb 1: TOV Roga ta koputa

Bulb 2: TOV Roga ta oleni

Bulb 3: TOV Koputa

Bulb 4: TOV Romashka

Bulb 5: TOV Kapysta

ВИСНОВКИ

При виконанні даної лабораторної роботи було набуто практичного досвіду у взаємодії з структуровані типами даних.