Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра «Мультимедійні та інтернет технології і системи»

**ЗВІТ**

про виконання лабораторної роботи №{{report\_number}}

«{{theme}}»

з навчальної дисципліни «Програмування ч.2»

**Варіант {{variant}}**

Виконав: студент групи {{group}}

{{student\_full\_name}}

Перевірив: доцент кафедри МІТС

Носик А.М.

Харків-{{year}}

**Мета лабораторної роботи**

text

**Завдання лабораторної роботи**

text

**Хід роботи**

{% for item in tasks%}

{{ item }}

{% endfor %}

**Завдання N**

#include <stdio.h>

**Висновки**

{{ conclusions }}

**Додаток з кодом**

**main.c**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include "task\_manager.h"

// Розкоментуйте строку, якщо компілюєте програму на Windows

// #define IS\_WINDOWS

#ifdef IS\_WINDOWS

#include <windows.h>

#endif

void init\_code\_page()

{

// Перевірка чи компілюється програма на Windows чи ні

#ifdef IS\_WINDOWS

SetConsoleOutputCP(1251);

#endif

}

int task1\_foo(char \*filename)

{

FILE \*file;

file = fopen(filename, "r");

if (file == NULL)

{

printf("Не вдалося відкрити файл '%s'\n", filename);

return 1;

}

char name[100];

fgets(name, sizeof(name), file);

printf("Інформація з файлу '%s':\n%s\n\n", filename, name);

fclose(file);

return 0;

}

void task1()

{

task1\_foo("input.txt");

task1\_foo("../input.txt");

task1\_foo("hmm.txt");

}

int task2()

{

FILE \*input\_file;

FILE \*output\_file;

int number\_of\_input\_files = 3;

char input\_files\_names[][20] = {"input1.txt", "input2.txt", "input3.txt"};

char output\_file\_name[] = "output.txt";

char data[100];

output\_file = fopen(output\_file\_name, "a");

if (output\_file == NULL)

{

printf("Не вдалося відкрити файл для запису '%s'\n", output\_file\_name);

return 1;

}

for (int i = 0; i < number\_of\_input\_files; i++)

{

input\_file = fopen(input\_files\_names[i], "r");

if (input\_file == NULL)

{

printf("Не вдалося відкрити вхідний файл '%s'\n", input\_files\_names[i]);

return 1;

}

fgets(data, sizeof(data), input\_file);

fputs(data, output\_file);

fclose(input\_file);

}

fclose(output\_file);

printf("Інформація з трьох вхідних файлів була успішно записана в '%s'\n", output\_file\_name);

return 0;

}

int task3\_fill(int n, char \*filename)

{

FILE \*file;

int \*numbers = malloc(sizeof(int) \* n);

srand(time(NULL));

for (int i = 0; i < n; i++)

{

numbers[i] = rand() % 90 + 10;

}

file = fopen(filename, "wb");

if (file == NULL)

{

printf("Не вдалося відкрити файл для запису '%s'\n", filename);

return 1;

}

fwrite(numbers, sizeof(int), n, file);

fclose(file);

return 0;

}

int task3\_print(int n, int variant\_number, char \*filename)

{

FILE \*file;

file = fopen(filename, "rb");

if (file == NULL)

{

printf("Не вдалося відкрити файл для читання '%s'\n", filename);

return 1;

}

printf("Вміст файлу '%s':\n", filename);

int \*numbers = malloc(sizeof(int) \* n);

fread(numbers, sizeof(int), n, file);

for (int i = 0; i < n; i++)

{

printf("%d ", numbers[i]);

}

printf("\n");

int position = (variant\_number - 1) % n;

printf("Число на позиції %d: %d\n", position, numbers[position]);

fclose(file);

return 0;

}

void task3()

{

const int n = 16;

char filename[] = "bin.dat";

int variant\_number = 7;

task3\_fill(n, filename);

task3\_print(n, variant\_number, filename);

return 0;

}

int main()

{

init\_code\_page();

struct TaskList task\_list = {NULL, 0};

register\_task(&task\_list, task1, 1, "Завдання 1");

register\_task(&task\_list, task2, 2, "Завдання 2");

register\_task(&task\_list, task3, 3, "Завдання 3");

run\_task\_manager(&task\_list);

return 0;

}

**task\_manager.h**

#ifndef TASK\_MANAGER\_H\_INCLUDED

#define TASK\_MANAGER\_H\_INCLUDED

struct Task

{

void (\*task\_function)();

int id;

char \*title;

};

struct TaskList

{

struct Task \*tasks;

int count;

};

void register\_task(struct TaskList \*task\_list, void (\*task\_function)(), int id, char \*title);

void run\_tusk(struct TaskList \*task\_list, int task\_number);

void run\_task\_manager(struct TaskList \*task\_list);

#endif // TASK\_MANAGER\_H\_INCLUDED

**task\_manager.c**

#include <stdio.h>

#include <stdbool.h>

#include "task\_manager.h"

void register\_task(struct TaskList \*task\_list, void (\*task\_function)(), int id, char \*title)

{

// Збільшення розміру масиву на один елемент

task\_list->count++;

task\_list->tasks = realloc(task\_list->tasks, task\_list->count \* sizeof(struct Task));

// Додавання нового завдання до масиву

task\_list->tasks[task\_list->count - 1].task\_function = task\_function;

task\_list->tasks[task\_list->count - 1].id = id;

task\_list->tasks[task\_list->count - 1].title = title;

}

void run\_tusk(struct TaskList \*task\_list, int task\_number)

{

bool is\_any\_task\_start = false;

for (int i = 0; i < task\_list->count; i++)

{

if (task\_list->tasks[i].id == task\_number)

{

is\_any\_task\_start = true;

printf("%s\n", task\_list->tasks[i].title);

printf("========================================\n\n");

task\_list->tasks[i].task\_function();

printf("\n========================================\n");

}

}

if (!is\_any\_task\_start)

{

printf("========================================\n");

printf("Не зайдено жодного завдання під номером %d\n", task\_number);

printf("========================================\n");

}

}

void run\_task\_manager(struct TaskList \*task\_list)

{

printf("Доступні опції:\n");

printf("%3d -- вийти з програми\n", -1);

for (int i = 0; i < task\_list->count; i++)

{

printf("%3d -- %s\n", task\_list->tasks[i].id, task\_list->tasks[i].title);

}

int task\_number;

while (true)

{

printf("\n\n");

printf("Введіть номер опціїї або номер завдання: ");

scanf("%d", &task\_number);

printf("\n");

if (task\_number == -1)

{

break;

}

run\_tusk(task\_list, task\_number);

};

}