НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Безпечне програмування»

*Звiт з лабораторної роботи №7*

*Тема: «Дослідження витоків пам’яті»*

Виконав:

ст. гр. 16Б

Башкатов В.Є.

Харків – 2019

***Мета:*** дослідити механізми витоків пам’ятів та засобів боротьби з ними.

***Завдання***

Написати програму, що демонструє витоки пам’яті.

- Порівняти аналіз витоків пам’яті утіліт valgrind, функції \_crtdumpmemoryleaks. Додати ще якнайменьш одну утілиту для аналізу витоків пам’яті. Зробити порівняльний аналіз та навести практичні рекомендації щодо використання цих утілит.

- переписати роботу з вказівниками на роботу з smart-вказівниками та довести, що їх використання сприяє зменшенню об’ємів витоків пам’яті. **Текст програми**

#define \_CRTDBG\_MAP\_ALLOC

#include <stdlib.h>

#include <crtdbg.h>

#include <vld.h>

void main()

{

char \*pointer = NULL;

for (int i = 0; i < 10; i++) {

pointer = new char[100];

}

delete[] pointer;

for (int i = 0; i < 10; i++)

{

std::unique\_ptr <char> mass[100];

}

\_CrtDumpMemoryLeaks();

}

**Хід роботи:**

В ході роботи було проведено дослідження, що демонструє витоки пам’яті. Виконуємо аналіз витоку пам’яті за допомогою \_CrtDumpMemoryLeaks() з звичайного покажчику (рис.1).

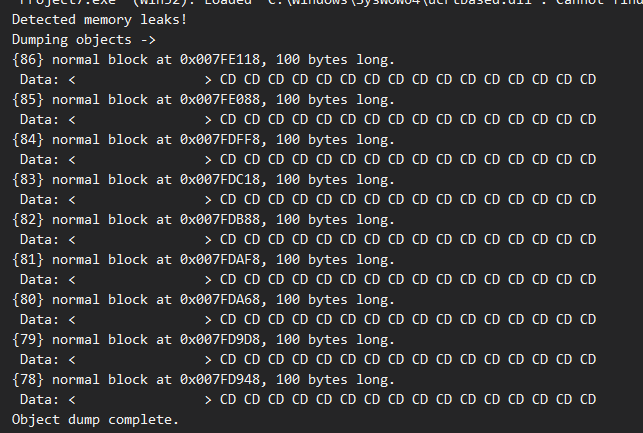


Рисунок 1 – Результат роботи програми з \_CrtDumpMemoryLeaks()

Та за допомогою бібліотеки Visual Leak Detector звичайного покажчику (<vld.h>) рис.2.

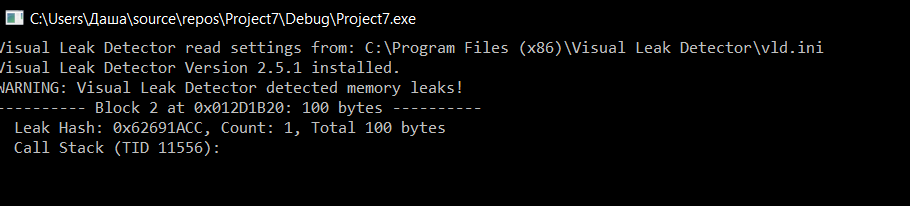


Рисунок 2 – Результат роботи програми з Visual Leak Detector

Результати роботи з розумним покажчиком з \_CrtDumpMemoryLeaks() та Visual Leak Detector(рис.3).

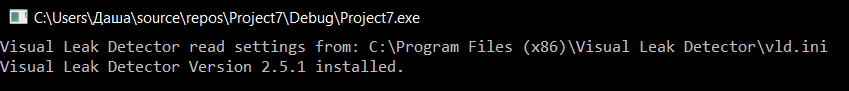


Рисунок 3 - – Результат роботи програми з Visual Leak Detector

Як ми бачимо, з результатів роботи цієї програми, що при використані розумного покажчику зменшує об’єми витоки пам’яті.

**Висновки:**

В результаті виконання лабораторної роботи були досліджені механізми витоків пам’ятів та засобів боротьби з ними.