Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра Обчислювальна техніка та програмування

Звіт з лабораторної роботи з дисципліні « Реверсне програмування »

Лабораторна робота 3

Виконав ст.гр. КІТ–36

Надірян Г.О.

Перевірив

Челак В.В.

**Харків 2019**

**Лабораторна робота 3**

**Тема**: Впровадження коду в ехе-файл в кінець секції з додаванням API-функцій

##### Мета: набути практичних навичок виправлення помилок в програмному забезпеченні, яке знаходиться в експлуатації створення нової функціональності, використовуючи неявне в експлуатації програмне забезпечення для платформи х64 в середовищі masm64.

##### Постановка задачі

Виконати завдання в кілька етапів:

1.В початку програми здійснити перевірку можливості мікропроцесора підтримувати команди AVX і AVX2. Про що вивести функціями MessageBox відповідні повідомлення.

2.Напісать програму в середовищі masm64 відповідно до варіанта завдання. Вивести 10 отриманих результатів через функцію MessageBox, а також номер варіанта, назва лабораторної роботи, шифр групи, прізвище виконавця. При використанні інструкцій доцільно скористатися довідниками, наприклад https://www.felixcloutier.com/x86/index.html.

3.Проізвесті підрахунок вільного місця в секції коду.

4.Осуществіть впровадження коду після виведення повідомлень про перевірку можливості мікропроцесора підтримувати команди AVX і AVX2 і перед виведенням спрощеного вікна з основними результатами програми.

Завдання на впровадження коду: впровадити додаткову функцію MessageBox з висновком слова «інфікованo», а так само шифр групи та прізвище виконавця (або стільки символів, скільки вміщається).

**Індивідуальне завдання**

Виконати дії над масивами (масивом) з 10 чисел і вивести отримані результати. Виконати інструкції VPMINSB.

**Код програми**

include win64a.inc

.data

a1 dd 0,1,1,2,3,5,8,13,21,34,55,89,144,14,15,16,17,18 ;18 чисел

b1 dd 1,2,3,4,6,6,7,8,9,10,11,12,13,14,16,17,18,19 ;18 чисел

len1 EQU ($-b1)/type b1

res1 dd 18 dup(0),0

res2 dq len1 dup(0),0

buf1 dq len1 dup(0),0

frmt db "Использование команды VPMINSB",10,

"0,1,1,2,3,5,8,13,21,34,55,89,144,14,15,16,17,18",10,

"1,2,3,4,6,6,7,8,9,10,11,12,13,14,16,17,18,19",10,10,

"MIN:", 10 dup(" %d "),10,10,0

.code

WinMain proc

push rbp ; <— это уже выравнивает стек на 8

sub rsp,30h ; <— для 7-10 параметров

mov rbp,rsp

.data

titl db "Проверка микропроцессора на поддержку команд AVX2",0 ; название окна

szInf db "Команды AVX ПОДДЕРЖИВАЮТСЯ!!!",0 ;

inf db "Команды AVX микропроцессором НЕ поддерживаются",0

.code

; проверка на поддержку AVX команд

mov EAX,1 ; при использования 64-разрядной ОС

cpuid ; по содержимому eax производится идентификация микропроцессора

and ecx,10000000h ; eсx:= eсx v 1000 0000h (28 разряд)

jnz exit1 ; перейти на exit, если не ноль

invoke MessageBox,0,addr inf,addr titl,MB\_OK

jmp exit2

exit1:

invoke MessageBox,0,addr szInf,addr titl,MB\_ICONINFORMATION

exit2:

.data

tit2 db "Проверка микропроцессора на поддержку команд AVX2",0 ; название окна

szInf2 db "Команды AVX2 ПОДДЕРЖИВАЮТСЯ!!!",0 ;

inf2 db "Команды AVX2 микропроцессором НЕ поддерживаются",0

.code

mov eax,7

mov ecx,0

cpuid ; по содержимому rax производится идентификация МП

and rbx,20h ; (5 разряд)

jnz @f ; перейти, если не нуль

invoke MessageBox,0,addr inf2,addr tit2,MB\_OK

@@:

invoke MessageBox,0,addr szInf2,addr tit2,MB\_ICONINFORMATION

mov rax,len1 ;

mov rbx,8 ; 32 х 8 = 256

xor rdx,rdx

div rbx ; определение количества циклов в rax и остатка в rdx

mov rcx,rax

lea rsi,a1

lea rdi,b1

lea rbx,res1

m1: vmovups ymm0,[rsi]

vmovups ymm1,[rdi]

VPMINSB ymm2,ymm0,ymm1

vmovups [rbx],ymm2

add rdi,32 ; 32 х 8 = 256

add rsi,32 ; смещение на 256

add rbx,32 ; смещение на 32 байта = 256

loop m1

cmp rdx,0h ; сравнение остатка с нулем

jz exit10 ; перейти, если ноль

mov rcx,rdx ; занесение содержимого rdx в счетчик

m2:

loop m2

exit10:

lea rsi,res1

lea rdi,res2

mov rcx,len1 ; количество чисел, которые выводятся в окно

m5:

movsxd r15,dword ptr [rsi]

mov qword ptr[rdi],r15

add rsi,4

add rdi,8

dec rcx

jnz m5

invoke wsprintf,ADDR buf1,ADDR frmt,res2,res2[8],res2[16],res2[24],res2[32],res2[40], res2[48],res2[56],res2[64],res2[72];

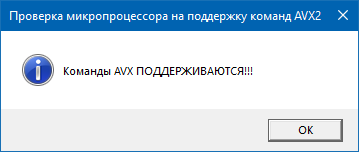
invoke MessageBox,0,ADDR buf1,ADDR titl,MB\_OK

invoke ExitProcess,0

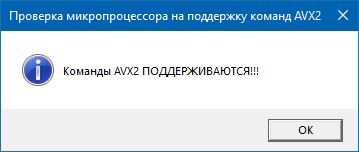
WinMain endp

End

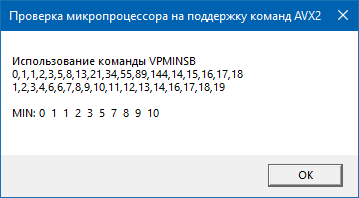
Работа програми до впровадження коду в ехе-файл в кінець секції з додаванням API-функцій



**Рисунок 1- Перевірка мікропроцесора на підтримку команд AVX**



**Рисунок 2- Перевірка мікропроцесора на підтримку команд AVX2**



**Рисунок 3 - Виконання індивідуального завдання**

**Визначення вільного місця в секції місця в секції .text**

Для підрахунку кількості байтів вільних осередків пам'яті необхідно з початкової адреси наступної секції відняти першу вільну комірку після закінчення коду 600h– 5B2h =4Eh

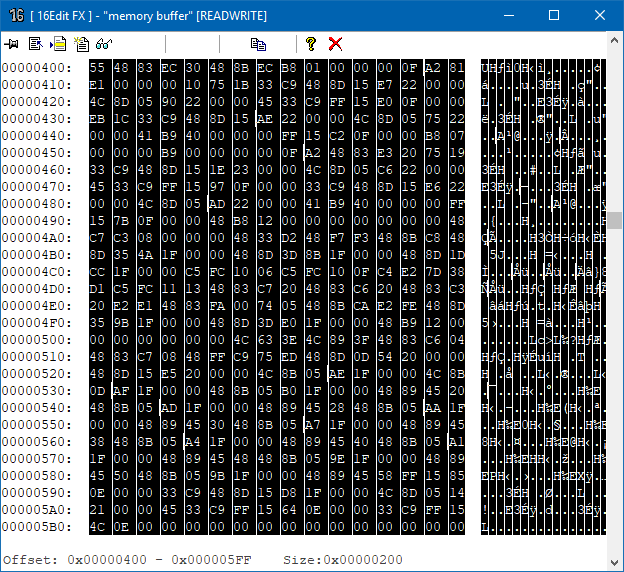


Рисунок 4 - **Визначення вільного місця в секції .text**

Спочатку встановлюємо курсор на першу командуі двічі натискаємо на ліву кнопку миші.(рис.2.)

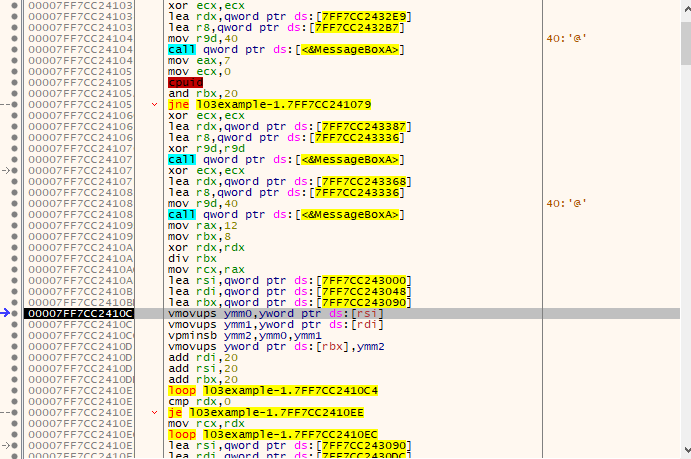


Рисунок 5 – Адреса першої команди

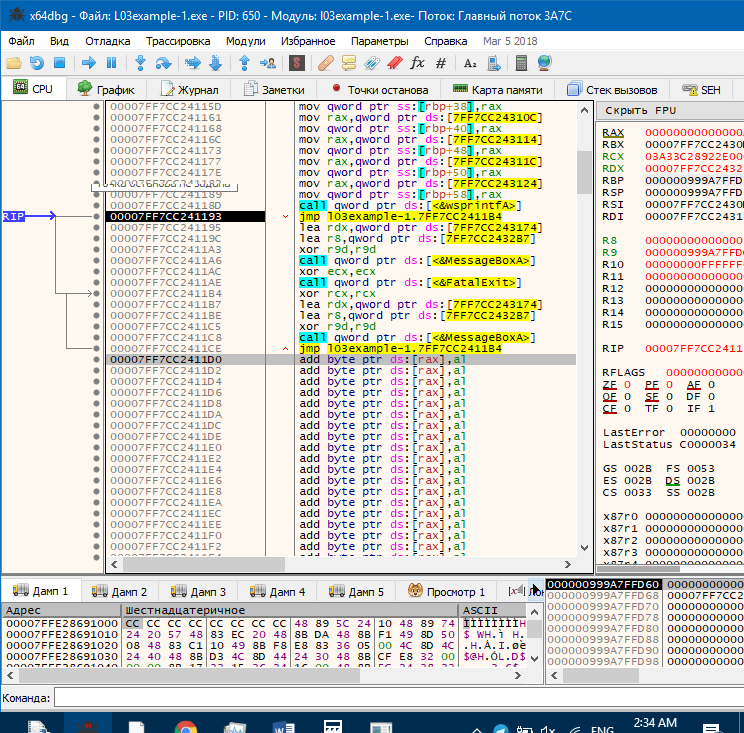


Рисунок 6 - Вставляємо параметри і викликаємо саму функцію MessageBox.

Міняємо адреси параметрів функції MessageBox та вставляємо в комірки пам'яті текст повідомлення

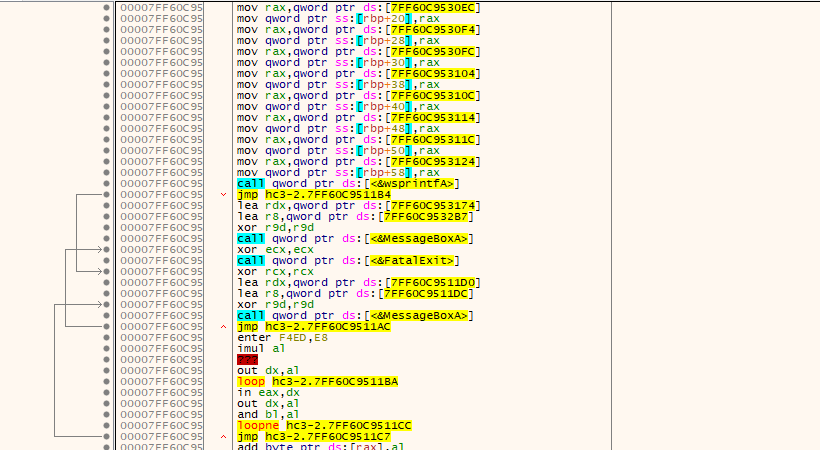
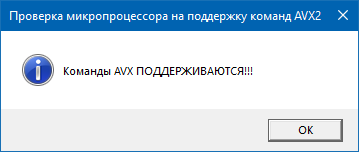
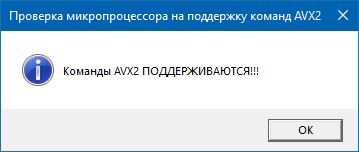


Рисунок 7 - Міняємо адреси параметрів функції MessageBox та вставляємо в комірки пам'яті текст повідомлення

Результат роботи після впровадження коду в ехе-файл в кінець секції з додаванням API-функцій



**Рисунок 8 - Перевірка мікропроцесора на підтримку команд AVX**



**Рисунок 9 - Перевірка мікропроцесора на підтримку команд AVX2**

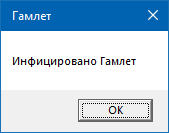


Рисунок 10 – Результат роботи після впровадження коду в ехе-файл в кінець секції з додаванням API-функцій

**Висновок**

##### Набути практичні навичок виправлення помилок в програмному забезпеченні, яке знаходиться в експлуатації створення нової функціональності, використовували неявне в експлуатації програмне забезпечення для платформи х64 в середовищі masm64. Впровадили код в ехе-файл в кінець секції з додаванням API-функцій.