НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №2

з дісципліни **«**Системне програмування**»**

Виконав:

студент 2 курсу

ФІОТ гр. ІО-32

Попенко Р. Л.

Перевірив:

Порєв В.М.

Київ 2015 р.

**Мета:** Отримати перші навички роботи з Microsoft Visual Studio для створення

програм, написаних мовою асемблера, а також вивчити команди MOV та

CPUID.

**Завдання:**

1. Створити у середовищі MS Visual Studio проект з ім’ям **Lab2**. Встановити

необхідні параметри проекту – опції середовища розробки програм.

2. Написати вихідний текст програми на асемблері, додати файл вихідного

тексту у проект. Зміст вихідного тексту згідно з варіантом завдання.

3. Скомпілювати вихідний текст і отримати виконуємий файл програми.

4. Перевірити роботу програми. Налагодити програму.

5. Отримати дизасембльований текст машинного коду і проаналізувати його.

Текст програми:

.586

.model flat, stdcall

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\user32.lib

.data

res dd 256 dup(0)

Text db 'EAX=xxxxxxxx',13,10,

'EBX=xxxxxxxx',13,10,

'ECX=xxxxxxxx',13,10,

'EDX=xxxxxxxx',0

Caption db "CPUID 0",0

Caption1 db "CPUID 1",0

Caption2 db "CPUID 2",0

Caption3 db "CPUID 80000000h",0

Caption4 db "CPUID ...01h",0

Caption5 db "CPUID ...02h",0

Caption6 db "CPUID ...03h",0

Caption7 db "CPUID ...04h",0

Caption8 db "CPUID ...05h",0

Caption9 db "CPUID ...06h",0

Caption10 db "CPUID ...07h",0

Caption11 db "CPUID ...08h",0

Vendor db 16 dup(0)

CaptionVendor db "CPUID 0 Vendor string",0

.code

DwordToStrHex proc

push ebp

mov ebp,esp

mov ebx,[ebp+8] ;другий параметр

mov edx,[ebp+12] ;перший параметр

xor eax,eax

mov edi,7

@next:

mov al,dl

and al,0Fh ;виділяємо одну шістнадцяткову цифру

add ax,48 ;так можна тільки для цифр 0-9

cmp ax,58

jl @store

add ax,7 ;для цифр A,B,C,D,E,F

@store:

mov [ebx+edi],al

shr edx,4

dec edi

cmp edi,0

jge @next

pop ebp

ret 8

DwordToStrHex endp

main:

mov eax, 0

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call DwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call DwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption, 0

mov eax, 0

cpuid

mov dword ptr[Vendor], ebx

mov dword ptr[Vendor+4], edx

mov dword ptr[Vendor+8], ecx

invoke MessageBoxA, 0, ADDR Vendor, ADDR CaptionVendor, 0

mov eax,1

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call DwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call DwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption1, 0

mov eax,2

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call DwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call DwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption2, 0

mov eax, 80000000h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call DwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call DwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption3, 0

mov eax, 80000001h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call DwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call DwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption4, 0

mov eax, 80000002h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call DwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call DwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption5, 0

mov eax, 80000003h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call DwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call DwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption6, 0

mov eax, 80000004h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call DwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call DwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption7, 0

mov eax, 80000005h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call DwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call DwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption8, 0

mov eax, 80000006h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call DwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call DwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption9, 0

mov eax, 80000007h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call DwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call DwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption10, 0

mov eax, 80000008h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call DwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call DwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

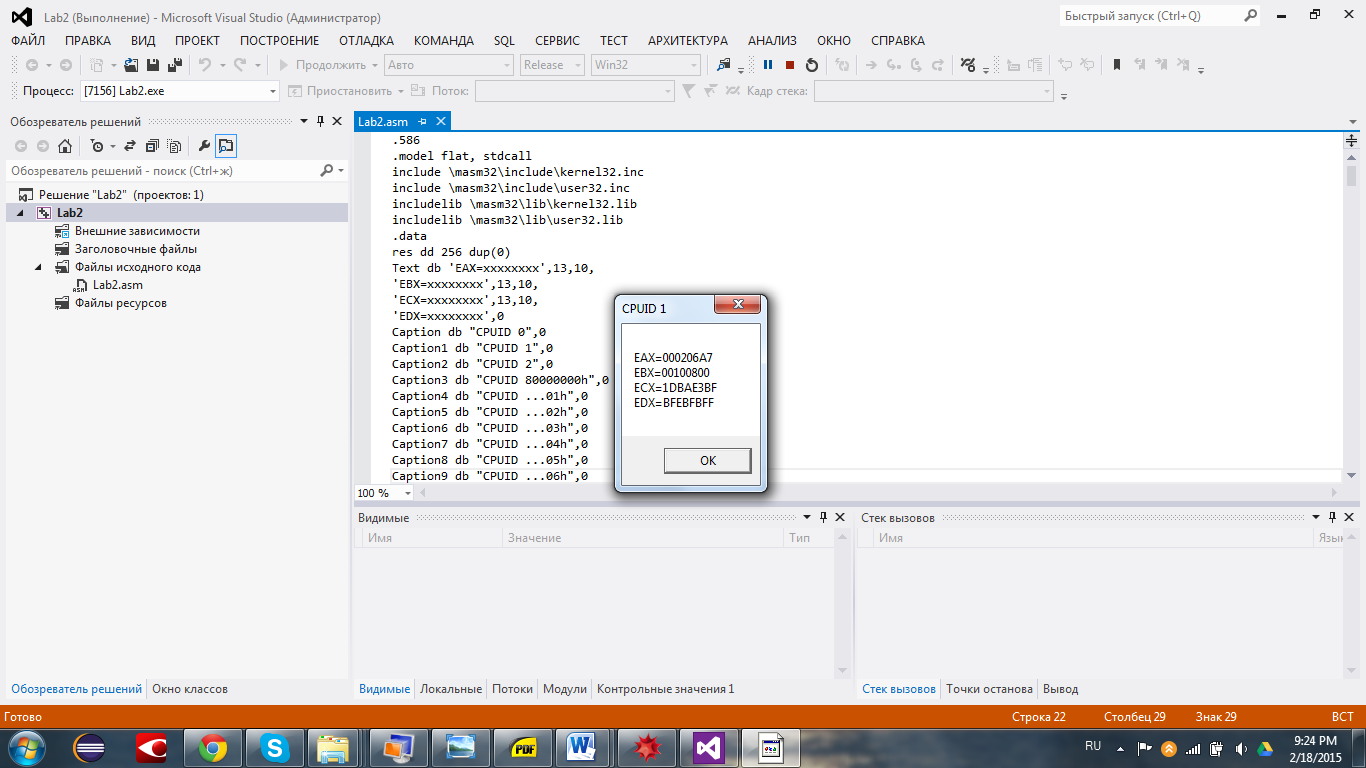
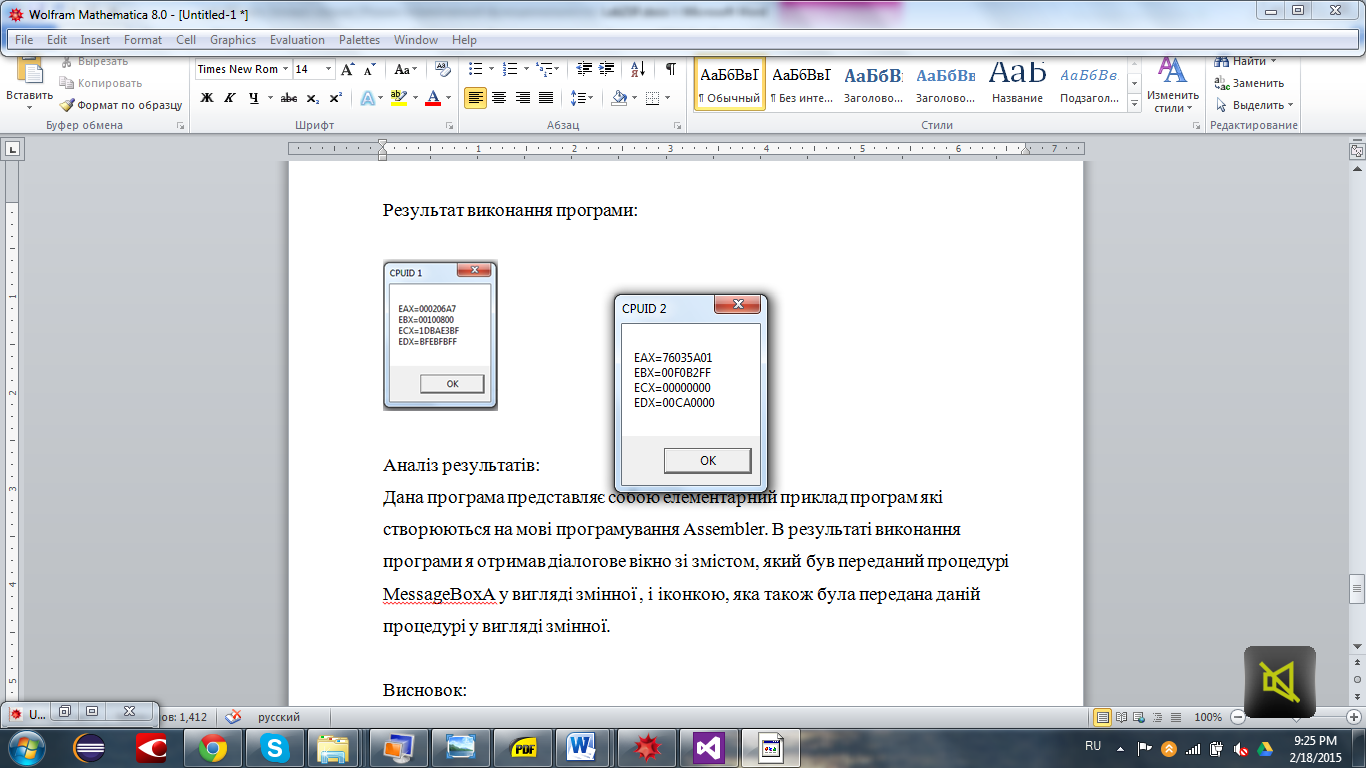
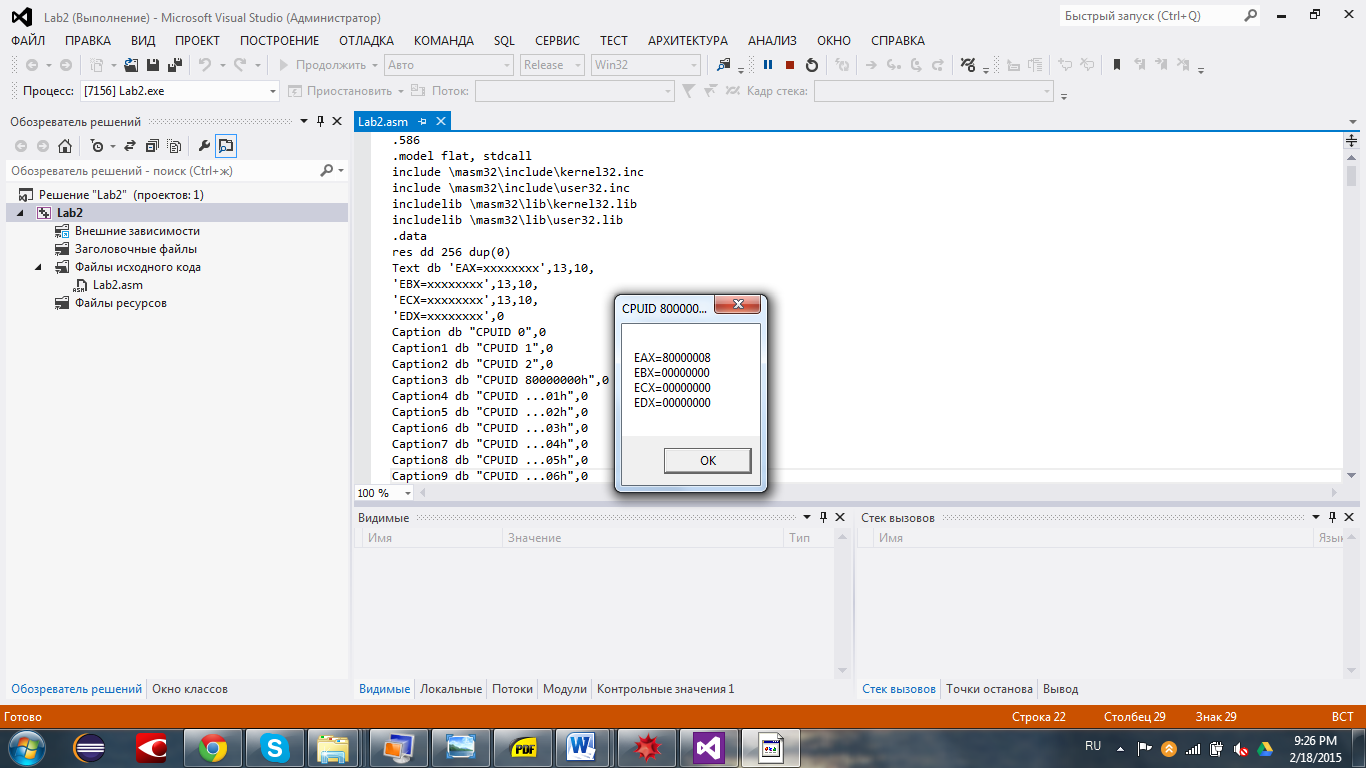
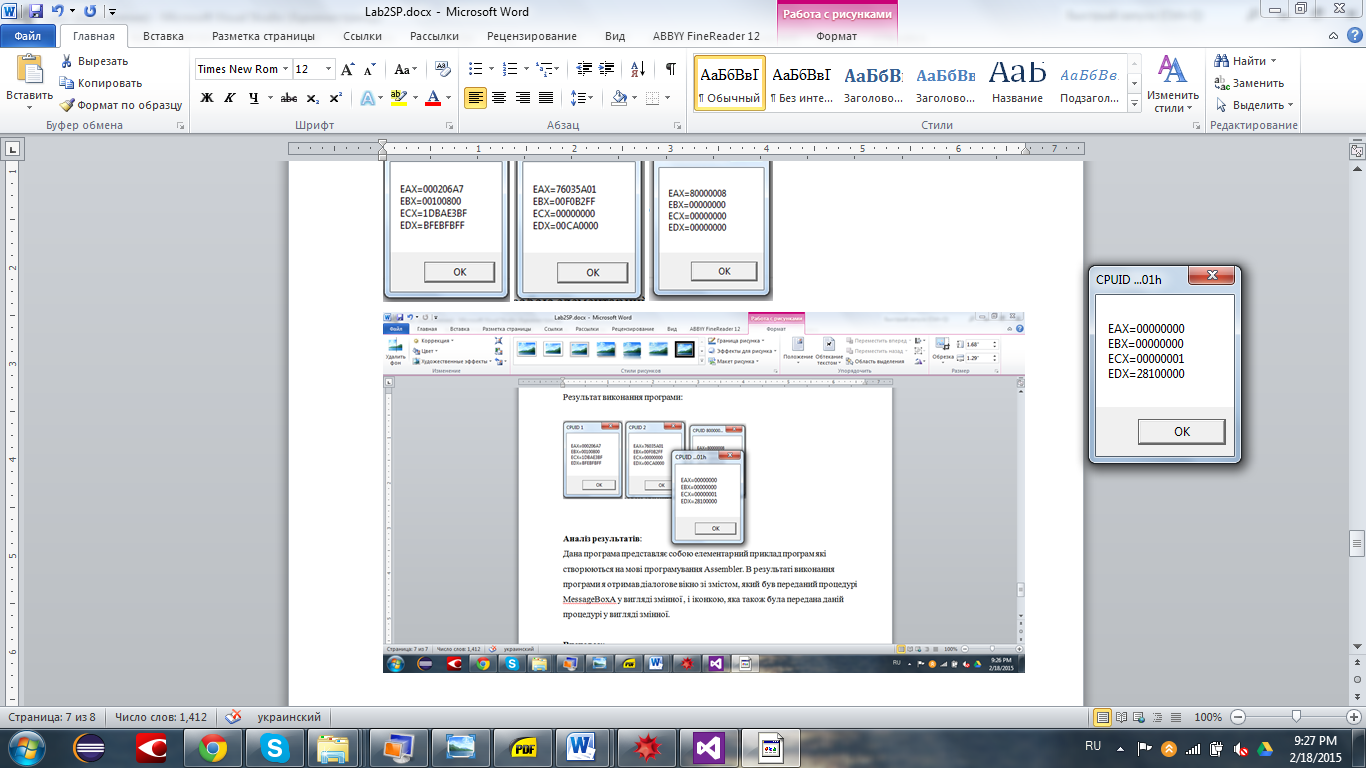
call DwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption11, 0

invoke ExitProcess, 0

end main

**Результат виконання програми**:

**Аналіз результатів**:

Дана програма представляє собою елементарний приклад програм які створюються на мові програмування Assembler. В результаті виконання програми я отримав діалогове вікно зі змістом, який був переданий процедурі MessageBoxA у вигляді змінної , і іконкою, яка також була передана даній процедурі у вигляді змінної.

**Висновок**:

Під час виконання даної лабораторної роботи мною був встановлений пакет Microsoft VisualStudio. Після завершення інсталяції був вивчений мною інтерфейс даного пакету, написана елементарна програма для перевірки працездатності даного пакету, а також програма яка відобразила інформацію про процесор мого комп’ютера. Під час написання програм були задіяні директиви invoke move, push і інші.