

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний Технічний Університет України
«Київський Політехнічний Інститут»
Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №2
з дисципліни «Системне програмування – 1»
на тему: «Знайомство із середовищем розробки програм Microsoft
Visual Studio»

Виконав:
студент 2-го курсу ФІОТ
групи ІВ-71
Мазан Я. В.

Перевірив:
Старший викладач
Порєв В. М.

Київ – 2019

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. Налогодити програму.
5. Отримати дизасембльований текст машинного коду і проаналізувати його.

Усім студентам необхідно запрограмувати:

- початкове діалогове вікно-вітання від автора програми;
- виконання команди CPUID з параметрами 0, 1, 2 а також 80000000h, 80000001h, 80000002h, 80000003h, 80000004h, 80000005h та 80000008h.

Кожний результат виконання CPUID команди потрібно виводити у окремому діалоговому вікні. Якщо результати CPUID утворюють текстові дані, то виводити їх як рядки тексту.

Код програми:

```
.586
.model flat, stdcall
include \masm32\include\kernel32.inc
include \masm32\include\user32.inc
includelib \lib\kernel32.lib
includelib \lib\user32.lib

.data
    GreetingCaption db "Лабораторна №2",0
    GreetingText db "Виконав Мазан Ян, ІВ-71",0
    InfoCaption0 db "При CPUID=0",0
    InfoCaption1 db "При CPUID=1",0
    InfoCaption2 db "При CPUID=2",0
    InfoCaption3 db "При CPUID=80000000h",0
    InfoCaption4 db "При CPUID=80000001h",0
    InfoCaption5 db "При CPUID=80000002h",0
    InfoCaption6 db "При CPUID=80000003h",0
    InfoCaption7 db "При CPUID=80000004h",0
    InfoCaption8 db "При CPUID=80000005h",0
    InfoCaption9 db "При CPUID=80000008h",0

    InfoText db "EAX=00000000",13,10,
               "EBX=00000000",13,10,
               "ECX=00000000",13,10,
               "EDX=00000000",0

    res dd 256 dup(0)

.code

;дя процедура записує 8 символів HEX коду числа
;перший параметр - 32-бітове число
;другий параметр - адреса буфера тексту
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:
invoke MessageBoxA, 0, ADDR GreetingText, ADDR GreetingCaption, 0

```

```

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]
push offset [InfoText+4]
call DwordToStrHex
push [res+4]
push offset [InfoText+18]
call DwordToStrHex
push [res+8]
push offset [InfoText+32]
call DwordToStrHex
push [res+12]
push offset [InfoText+46]
call DwordToStrHex
invoke MessageBoxA, 0, ADDR InfoText, ADDR InfoCaption0, 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]
push offset [InfoText+4]
call DwordToStrHex
push [res+4]
push offset [InfoText+18]
call DwordToStrHex
push [res+8]
push offset [InfoText+32]
call DwordToStrHex
push [res+12]
push offset [InfoText+46]
call DwordToStrHex
invoke MessageBoxA, 0, ADDR InfoText, ADDR InfoCaption1, 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]
push offset [InfoText+4]
call DwordToStrHex
push [res+4]
push offset [InfoText+18]
call DwordToStrHex
push [res+8]
push offset [InfoText+32]

```

```
call DwordToStrHex
push [res+12]
push offset [InfoText+46]
call DwordToStrHex
invoke MessageBoxA, 0, ADDR InfoText, ADDR InfoCaption2, 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]
push offset [InfoText+4]
call DwordToStrHex
push [res+4]
push offset [InfoText+18]
call DwordToStrHex
push [res+8]
push offset [InfoText+32]
call DwordToStrHex
push [res+12]
push offset [InfoText+46]
call DwordToStrHex
invoke MessageBoxA, 0, ADDR InfoText, ADDR InfoCaption3, 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]
push offset [InfoText+4]
call DwordToStrHex
push [res+4]
push offset [InfoText+18]
call DwordToStrHex
push [res+8]
push offset [InfoText+32]
call DwordToStrHex
push [res+12]
push offset [InfoText+46]
call DwordToStrHex
invoke MessageBoxA, 0, ADDR InfoText, ADDR InfoCaption4, 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]
push offset [InfoText+4]
call DwordToStrHex
push [res+4]
push offset [InfoText+18]
call DwordToStrHex
push [res+8]
push offset [InfoText+32]
call DwordToStrHex
push [res+12]
push offset [InfoText+46]
call DwordToStrHex
invoke MessageBoxA, 0, ADDR InfoText, ADDR InfoCaption5, 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]
push offset [InfoText+4]
call DwordToStrHex
push [res+4]
push offset [InfoText+18]
call DwordToStrHex
push [res+8]
push offset [InfoText+32]
call DwordToStrHex
push [res+12]
push offset [InfoText+46]
call DwordToStrHex
invoke MessageBoxA, 0, ADDR InfoText, ADDR InfoCaption6, 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]
push offset [InfoText+4]
call DwordToStrHex
push [res+4]
push offset [InfoText+18]
call DwordToStrHex
push [res+8]
push offset [InfoText+32]
call DwordToStrHex
push [res+12]
push offset [InfoText+46]
call DwordToStrHex
invoke MessageBoxA, 0, ADDR InfoText, ADDR InfoCaption7, 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
```

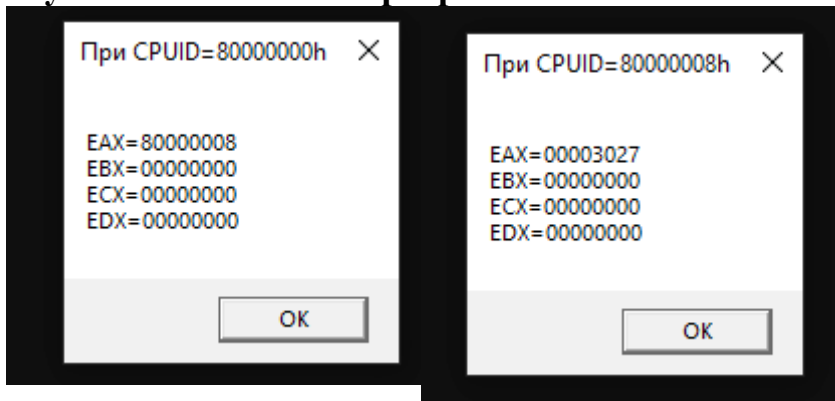
```
push [res]
push offset [InfoText+4]
call DwordToStrHex
push [res+4]
push offset [InfoText+18]
call DwordToStrHex
push [res+8]
push offset [InfoText+32]
call DwordToStrHex
push [res+12]
push offset [InfoText+46]
call DwordToStrHex
invoke MessageBoxA, 0, ADDR InfoText, ADDR InfoCaption8, 0
```

```
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]
push offset [InfoText+4]
call DwordToStrHex
push [res+4]
push offset [InfoText+18]
call DwordToStrHex
push [res+8]
push offset [InfoText+32]
call DwordToStrHex
push [res+12]
push offset [InfoText+46]
call DwordToStrHex
invoke MessageBoxA, 0, ADDR InfoText, ADDR InfoCaption9, 0
```

```
invoke ExitProcess, 0  
end main
```

Результат виконання програми:



Висновок:

Під час виконання даної лабораторної роботи мною був встановлений пакет Microsoft VisualStudio. Після завершення інсталяції був вивчений інтерфейс даного пакету, написана елементарна програма для перевірки працездатності даного пакету, а також програма, що відобразила інформацію про процесор мого комп'ютера. Мною були отримані перші навички роботи із засобами для створення програм, написаних мовою асемблера, а також вивчені команди MOV та CPUID.