Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №2**

з дисципліни «Системне програмування»

Виконав:

студент гр. ІО-33

Шуркіна Анастасія

Перевірив:

Порєв В. М.

Київ 2015

**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 виконуваний файл програми.

4. Перевірити роботу програми. Налагодити програму.

5. Отримати дизасембльований текст машинного коду і проаналізувати його.

**Текст програми:**

.model flat, stdcall

option casemap :none

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

include \masm32\include\windows.inc

includelib \masm32\\lib\kernel32.lib

includelib \masm32\lib\user32.lib

.data

Caption db 'Лабораторна робота №2', 0

Text db 'Гарного дня!', 13, 10, 'Шуркіна Анастасія',0

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

'EBX=xxxxxxxx',13,10,

'ECX=xxxxxxxx',13,10,

'EDX=xxxxxxxx',0

res dd 256 dup(0)

Caption0 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

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

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+16], eax

mov dword ptr[res+20], ebx

mov dword ptr[res+24], ecx

mov dword ptr[res+28], edx

mov eax, 2

cpuid

mov dword ptr[res+32], eax

mov dword ptr[res+36], ebx

mov dword ptr[res+40], ecx

mov dword ptr[res+44], edx

mov eax, 80000000h

cpuid

mov dword ptr[res+48], eax

mov dword ptr[res+52], ebx

mov dword ptr[res+56], ecx

mov dword ptr[res+60], edx

mov eax, 80000001h

cpuid

mov dword ptr[res+64], eax

mov dword ptr[res+68], ebx

mov dword ptr[res+72], ecx

mov dword ptr[res+76], edx

mov eax, 80000002h

cpuid

mov dword ptr[res+80], eax

mov dword ptr[res+84], ebx

mov dword ptr[res+88], ecx

mov dword ptr[res+92], edx

mov eax, 80000003h

cpuid

mov dword ptr[res+96], eax

mov dword ptr[res+100], ebx

mov dword ptr[res+104], ecx

mov dword ptr[res+108], edx

mov eax, 80000004h

cpuid

mov dword ptr[res+112], eax

mov dword ptr[res+116], ebx

mov dword ptr[res+120], ecx

mov dword ptr[res+124], edx

mov eax, 80000005h

cpuid

mov dword ptr[res+128], eax

mov dword ptr[res+132], ebx

mov dword ptr[res+136], ecx

mov dword ptr[res+140], edx

mov eax, 80000006h

cpuid

mov dword ptr[res+144], eax

mov dword ptr[res+148], ebx

mov dword ptr[res+152], ecx

mov dword ptr[res+156], edx

mov eax, 80000007h

cpuid

mov dword ptr[res+160], eax

mov dword ptr[res+164], ebx

mov dword ptr[res+168], ecx

mov dword ptr[res+172], edx

mov eax, 80000008h

cpuid

mov dword ptr[res+176], eax

mov dword ptr[res+180], ebx

mov dword ptr[res+184], ecx

mov dword ptr[res+188], edx

;next part

push [res]

push offset [Txt+4]

call DwordToStrHex

push [res+4]

push offset [Txt+18]

call DwordToStrHex

push [res+8]

push offset [Txt+32]

call DwordToStrHex

push [res+12]

push offset [Txt+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Txt, ADDR Caption0, 0

push [res+16]

push offset [Txt+4]

call DwordToStrHex

push [res+20]

push offset [Txt+18]

call DwordToStrHex

push [res+24]

push offset [Txt+32]

call DwordToStrHex

push [res+28]

push offset [Txt+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Txt, ADDR Caption1, 0

push [res+32]

push offset [Txt+4]

call DwordToStrHex

push [res+36]

push offset [Txt+18]

call DwordToStrHex

push [res+40]

push offset [Txt+32]

call DwordToStrHex

push [res+44]

push offset [Txt+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Txt, ADDR Caption2, 0

push [res+48]

push offset [Txt+4]

call DwordToStrHex

push [res+52]

push offset [Txt+18]

call DwordToStrHex

push [res+56]

push offset [Txt+32]

call DwordToStrHex

push [res+60]

push offset [Txt+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Txt, ADDR Caption3, 0

push [res+64]

push offset [Txt+4]

call DwordToStrHex

push [res+68]

push offset [Txt+18]

call DwordToStrHex

push [res+72]

push offset [Txt+32]

call DwordToStrHex

push [res+76]

push offset [Txt+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Txt, ADDR Caption4, 0

push [res+80]

push offset [Txt+4]

call DwordToStrHex

push [res+84]

push offset [Txt+18]

call DwordToStrHex

push [res+88]

push offset [Txt+32]

call DwordToStrHex

push [res+92]

push offset [Txt+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Txt, ADDR Caption5, 0

push [res+96]

push offset [Txt+4]

call DwordToStrHex

push [res+100]

push offset [Txt+18]

call DwordToStrHex

push [res+104]

push offset [Txt+32]

call DwordToStrHex

push [res+108]

push offset [Txt+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Txt, ADDR Caption6, 0

push [res+112]

push offset [Txt+4]

call DwordToStrHex

push [res+116]

push offset [Txt+18]

call DwordToStrHex

push [res+120]

push offset [Txt+32]

call DwordToStrHex

push [res+124]

push offset [Txt+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Txt, ADDR Caption7, 0

push [res+128]

push offset [Txt+4]

call DwordToStrHex

push [res+132]

push offset [Txt+18]

call DwordToStrHex

push [res+136]

push offset [Txt+32]

call DwordToStrHex

push [res+140]

push offset [Txt+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Txt, ADDR Caption8, 0

push [res+144]

push offset [Txt+4]

call DwordToStrHex

push [res+148]

push offset [Txt+18]

call DwordToStrHex

push [res+152]

push offset [Txt+32]

call DwordToStrHex

push [res+156]

push offset [Txt+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Txt, ADDR Caption9, 0

push [res+160]

push offset [Txt+4]

call DwordToStrHex

push [res+164]

push offset [Txt+18]

call DwordToStrHex

push [res+168]

push offset [Txt+32]

call DwordToStrHex

push [res+172]

push offset [Txt+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Txt, ADDR Caption10, 0

push [res+176]

push offset [Txt+4]

call DwordToStrHex

push [res+180]

push offset [Txt+18]

call DwordToStrHex

push [res+184]

push offset [Txt+32]

call DwordToStrHex

push [res+188]

push offset [Txt+46]

call DwordToStrHex

invoke MessageBoxA, 0, ADDR Txt, ADDR Caption11, 0

invoke MessageBoxA, 0, ADDR Text, ADDR Caption, MB\_OKCANCEL

invoke ExitProcess, 0

;ця 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

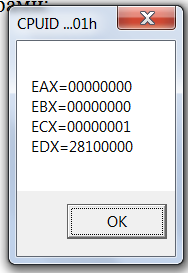
pop ebp

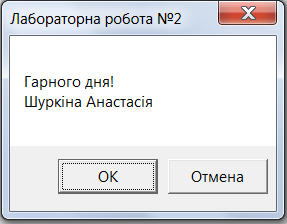
ret 8

DwordToStrHex endp

end main

Результати виконання програми:





**Аналіз результатів:**

Дана програма представляє собою елементарний приклад програм які створюються на мові програмування Assembler. В результаті виконання програми було отримано діалогові вікна зі змістом, який був переданий процедурі MessageBoxA у вигляді змінної .

**Висновок:**

При виконанні даної лабораторної роботи було встановлено пакет Microsoft VisualStudio. Після завершення інсталяції було вивчено 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 команди MOV, CPUID, використана процедура DwordToStrHex.